



СУПЕРКОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ



**Материалы 3-й Всероссийской
научно-технической конференции**

29 сентября - 4 октября 2014
Дивноморское, Геленджик

ТОМ 2

Министерство образования и науки Российской Федерации
Российский фонд фундаментальных исследований
Южный научный центр Российской академии наук
Южный федеральный университет
НИИ многопроцессорных вычислительных систем
имени академика А.В. Каляева Южного федерального университета
ОАО «НИЦЭВТ»
ФГУП «НИИ «Квант»
ООО «НИЦ супер-ЭВМ и нейрокомпьютеров»
Журнал «Вестник компьютерных и информационных технологий»
Журнал «Известия ЮФУ. Технические науки»

СУПЕРКОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

СКТ-2014

**Материалы 3-й Всероссийской
научно-технической конференции**

29 сентября – 4 октября 2014
Дивноморское, Геленджик

ТОМ 2

Ростов-на-Дону
2014

УДК 004.272.43
ББК 32.973
С 73

С 73 Суперкомпьютерные технологии (СКТ-2014) // Материалы 3-й Всероссийской научно-технической конференции: в 2 т. – Ростов-на-Дону. Издательство Южного федерального университета, 2014. – 272 с.
ISBN 978-5-9275-1282-9
ISBN 978-5-9275-1284-3 (Т.2)

Во втором томе материалов 3-й Всероссийской научно-технической конференции «Суперкомпьютерные технологии» (СКТ-2014) представлены доклады по направлениям: проблемно-ориентированные и реконфигурируемые вычислительные системы, распределенные вычислительные и управляющие системы, применение суперкомпьютерных технологий в науке, технике и промышленности.

Издание осуществлено при финансовой поддержке Российского фонда фундаментальных исследований, проект № 14-07-20307 г

М $\frac{2404000000}{6КО(03) - 2014}$ без объявл.

ISBN 978-5-9275-1284-3 (Т.2)
ISBN 978-5-9275-1282-9

УДК 004.272.43
ББК 32.973

© Научно-исследовательский институт многопроцессорных вычислительных систем имени академика А.В. Каляева федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет», составление, оформление, 2014

СОДЕРЖАНИЕ

РАЗДЕЛ 3. ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЕ И РЕКОНФИГУРИРУЕМЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Афонин А.А.

Малогабаритный микроконтроллерный вычислительный модуль для комплекса систем ориентации, навигации, гравиметрии и управления аэроморских динамичных объектов 9

Бовкун А.В.

Оптимизация аппаратного ресурса прикладной задачи за счет архитектурных особенностей кристаллов ПЛИС 14

Бугаев С.А., Завтур Е.Е., Шелестенко Е.Ю.

Адаптация алгоритма цифрового формирования квадратурных составляющих для реализации на ПЛИС 18

Елизаров Г.С., Горбунов В.С., Малахов И.Н., Титов А.Г.

Компоненты высокопроизводительных реконфигурируемых суперкомпьютеров на основе ПЛИС Xilinx Ultrascale 21

Зачесов Ю.Л., Золотов И.Н.

Исследование возможности достижения эксафлопсной производительности для одной трудоёмкой задачи 25

Катаев О.В.

Организация вычислительной системы для предварительной обработки тепловизионных сигналов 29

Колодзей А.В.

Компромисс «время/память» в реконфигурируемых вычислительных системах 32

Коротков К.А.

Защищенная компьютерная сеть реконфигурируемой вычислительной системы 35

Левин И.И., Дордопуло А.И., Каляев И.А., Семерников Е.А.

Вычислительные системы с реконфигурируемой архитектурой на основе ПЛИС Virtex-7 38

Лукин Н.А.

Специализация вычислений и функционально-ориентированные процессоры для встроенных суперкомпьютеров 46

Михайлин Д.А., Тучин Д.А., Фитенко В.В., Ярошевский В.С.

Математическая модель доплеровского измерителя скорости и дальности 54

Пелипец А.В., Коваленко А.Г.

Решение задачи факторизации больших чисел методом эллиптических кривых на реконфигурируемых вычислительных системах 55

Степаненко С.А.

Оценки ускорения вычислений гибридными реконфигурируемыми системами 58

Талдыкин Е.В., Телегин М.В., Непочатых Е.В.

Применение языка OpenCL при разработке систем на кристалле в FPGA фирмы Altera 63

Талдыкин Е.В., Телегин М.В., Непочатых Е.В.

Реконфигурируемый вычислительный модуль на базе ПЛИС CycloneV 67

Тарасов И.Е., Бялик Ю.И., Кузелин М.О.

Особенности системной архитектуры семейства FPGA Xilinx UltraScale 70

Угольков А.В., Шелестенко Е.Ю.

Реализации цифрового рекурсивного фильтра с использованием ПЛИС 76

РАЗДЕЛ 4. РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ И УПРАВЛЯЮЩИЕ СИСТЕМЫ

Бычков И.В., Опарин Г.А., Феоктистов А.Г.,

Богданова В.Г., Пашинин А.А.

Мультиагентное управление распределенными вычислениями на основе сервис-ориентированного подхода 80

Валуев И.А., Морозов И.В.

GridMD: компактная переносимая библиотека C++ для управления распределенными вычислениями 85

Вятчин Д.А., Доморацкий А.В., Николаева Е.Д.

Обучение систем обнаружения вторжений для распределенных
вычислительных систем..... 90

Голиков А.Н.

Алгоритм приближения функций на основе поэкстремумного
разбиения отрезка интерполяции с естественным параллелизмом..... 94

Горелова Г.В., Мельник Э.В., Клименко А.Б.

Формализация задачи формирования конфигурации информаци-
онно-управляющих систем с учетом параметра загрузки процес-
сорных устройств..... 99

Дацюк О.В., Дацюк В.Н., Крукиер Л.А.

Построение простейшей ГРИД-системы с использованием Globus
Toolkit 104

Егерева И.А., Палюх Б.В., Семухина М.С.

Задача систематизации научно-вычислительных сервисов рас-
пределенной среды 109

Иванов Д.Я.

Распределенное управление большой группой роботов при реше-
нии строевой задачи 114

Каляев А.И.

Методы повышения надежности систем передачи данных в кор-
поративных вычислительных сетях..... 118

Клименко А.Б.

Экспериментальный анализ эффективности имитации отжига
при решении задачи формирования конфигурации информаци-
онно-управляющей системы..... 121

Коробкин В.В., Серогодский А.И.

Оценка безопасности программного обеспечения распределен-
ных вычислительных и управляющих систем..... 125

Куповых В.Г.

Распределенные системы управления для высокорисковых производств..... 129

Лабошин Л.Ю., Лукашин А.А.

Подход к анализу сверхбольших снимков сетевого трафика с по-
мощью Map/Reduce..... 132

Машошин А.И., Соколов А.И.

Интегрированная система управления автономного необитаемого подводного аппарата 136

Мельник Э.В., Иванов Д.Я.

Надежность сетцентрических информационно-управляющих систем при распределенном диспетчировании 141

Мельник Э.В., Таранов А.Ю., Ольшанский М.Ю., Альбицкий Д.В.

К вопросу об организации реконфигурации информационно-управляющих систем интегрированной модульной авионики с использованием алгоритма имитации отжига 145

Меркулов В.И., Миляков Д.А., Самодов И.О.

Оптимизация алгоритма группового управления беспилотными летательными аппаратами в составе локальной сети 149

Павский В.А., Павский К.В.

Расчет показателей осуществимости решения задач на распределенных вычислительных системах с накопителем 152

Пестов Д.А.

Принципы построения распределенного информационно-измерительного комплекса автоматизированного мониторинга атмосферно-электрических параметров приземного слоя атмосферы 156

Сузанский Д.Н., Иванова О.А.

Задача управления полетом беспилотных летательных аппаратов в групповом порядке 160

Трегубов Д.Е.

Распределенная система контроля и управления технологическим процессом переработки радиоактивных отходов 162

Шляхов А.С., Сесекин А.Н.

Оптимизация формирования строя группой беспилотных летательных аппаратов – квадрокоптеров 168

РАЗДЕЛ 5. ПРИМЕНЕНИЕ СУПЕРКОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ В НАУКЕ, ТЕХНИКЕ И ПРОМЫШЛЕННОСТИ

Бахтерев М.О., Васёв П.А.

Один метод визуализации результатов суперкомпьютерных вычислений 172

Безруков А.В.

Построение эффективной ИТ-инфраструктуры для инженерных расчетов... 177

Боев С.Ф., Бкайтин А.В., Кочкаров А.А., Ступин Д.Д.

Проблемы и методы обработки интенсивных потоков данных в перспективных национальных системах мониторинга сложных процессов и многопараметрических сред..... 181

Буза М.К., Кондратьева О.М.

Использование суперкомпьютерных технологий для моделирования процессов в физике полупроводников..... 186

Бухановский А.В., Васильев В.Н., Нечаев Ю.И.

Виртуальный полигон – центр компетенции на основе современной теории катастроф..... 191

Бухановский А.В., Иванов С.В., Ковальчук С.В., Нечаев Ю.И.

Программный комплекс «Виртуальный полигон – центр компетенции», обеспечивающий оперативный контроль чрезвычайных ситуаций в транспортной и социальной среде на основе облачных вычислений и ГРИД-технологий 197

Ганченко В.В., Воронов А.А., Дудкин А.А., Инютин А.В.

Параллельная обработка изображений в производстве интегральных микросхем..... 203

Ганченко В.В., Дудкин А.А., Петровский А.И.

Параллельная подсистема для мониторинга состояния лесных угодий 208

Гурьев Ю.В., Ткаченко И.В., Якушенко Е.И.

Использование высокопроизводительных вычислений в задачах корабельной гидродинамики..... 212

Дворкович В.П., Дворкович А.В.

Новые методы цифровой обработки и спектрального анализа динамических изображений сверхвысокой четкости..... 217

Змеев Д.Н., Климов А.В., Левченко Н.Н., Окунев А.С.

Применение параллельной потоковой вычислительной системы «Буран» для задач молекулярной динамики..... 222

Котляров А.С.

Синтез конфигурационных файлов на кластерных вычислительных системах..... 225

Крукиер Л.А., Чикин А.Л., Чикина Л.Г.

Использование многопроцессорных вычислительных систем при моделировании наводнения в дельте Дона 226

Лайком Д.Н., Аксенов С.В.

Модификация алгоритма извлечения текстурных характеристик с помощью вычислений на графических процессорах 231

Маркович И.И., Копытин А.С., Марьев А.А.

Цифровая обработка сигналов в радиолокационном комплексе, использующем зондирующие сигналы с линейной частотной модуляцией и изменяющимся знаком девиации частоты 235

Михеев В.А., Изгалин С.П., Симонов А.С.

Применение суперкомпьютерных технологий разработки ОАО «НИЦЭВТ» в прикладных системах 240

Мурашко Н.И., Орешкина Л.В.

Параллельная обработка разновременных космических снимков 242

Петров И.Б., Хохлов Н.И.

Моделирование сейсмических процессов на высокопроизводительных вычислительных системах 246

Путилин А.Б., Степанов М.В., Соколова Ж.В.

Массивно параллельный нейросетевой алгоритм на базе CUDA в нанокompиляторе для оптимизации топологии 2D/3D наносхем на базе квантовых клеточных автоматов 250

Чекина М.Д.

Математическое моделирование задач геофильтрации в почвогрунтах с фрактальной структурой на многопроцессорных вычислительных системах 258

Шабас И.Н.

Расчет на высокопроизводительных вычислительных системах распространения в водоеме нефтяного пятна 263

Шишени А.В.

Параллельное численное моделирование задач гидродинамики мелководных водоемов с уточненными граничными условиями 266

АВТОРСКИЙ УКАЗАТЕЛЬ 270

РАЗДЕЛ 3

Проблемно-ориентированные и реконфигурируемые вычислительные системы

А.А. Афонин

МАЛОГАБАРИТНЫЙ МИКРОКОНТРОЛЛЕРНЫЙ ВЫЧИСЛИТЕЛЬНЫЙ МОДУЛЬ ДЛЯ КОМПЛЕКСА СИСТЕМ ОРИЕНТАЦИИ, НАВИГАЦИИ, ГРАВИМЕТРИИ И УПРАВЛЕНИЯ АЭРОМОРСКИХ ДИНАМИЧНЫХ ОБЪЕКТОВ*

*Московский авиационный институт (национальный
исследовательский университет), г. Москва,
kaf305-mai@mail.ru*

В настоящее время сохраняет свою актуальность проблема высокоточного определения параметров гравитационного поля Земли, в особенности в ее отдалённых и труднодоступных районах, акваториях морей и океанов, их шельфов. Традиционная аэроморская гравиметрия отличается отработанностью применяемых технологий и достаточным уровнем точности, при этом для нее характерны низкие производительность, детальность и оперативность, высокие материальные затраты, в особенности определяемые стоимостью эксплуатации среднего класса судов и ЛА с экипажем. В то же время, решение проблемы возможно посредством применения бортового миниатюрного управляющего бесплатформенного гравиинерциального навигационного комплекса (БГНК) малоразмерного динамичного носителя – беспилотного летательного (или подводного) аппарата (БЛА) [1].

В зависимости от целей применения и условий эксплуатации БГНК БЛА может существенно менять свой состав, однако общая структура и базовый набор основных узлов остаются неизменными. На рис. 1 представлена блок-схема рационального варианта БГНК с набором основных измерительных подсистем, к которым относятся: инерциальный измерительный блок, содержащий блок акселерометров (БА) и блок гироскопов (БГ), навигационная аппаратура потребителя спутниковой навигационной системы (СНС), магнитометрическая курсовая система (МКС) и система воздушных сигналов (СВС), включающая датчики воздушной скорости, углов атаки и скольжения.

* Работа выполнена при финансовой поддержке Минобрнауки РФ в рамках Госзадания по проекту № 813, 2014 г., рук. Афонин А.А.

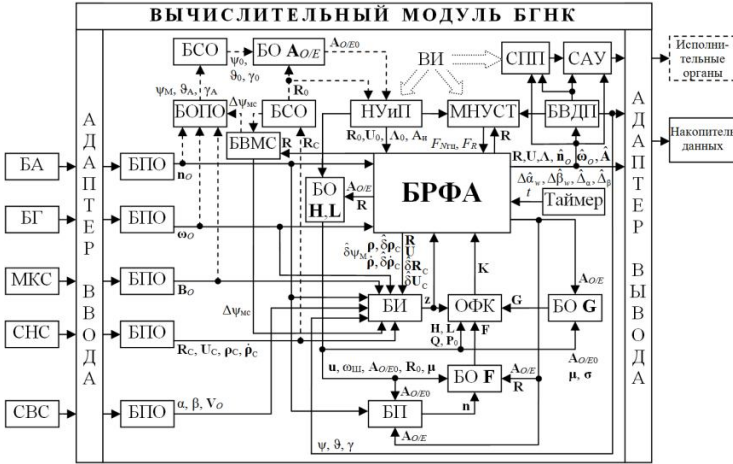


Рис. 1. Блок-схема БГНК

В рамках вычислительного модуля БГНК реализуется его программно-алгоритмическое обеспечение, включая алгоритмы ориентации, навигации, гравиметрии, оценивания параметров и коррекции, алгоритмы системы планирования полета (СПП), законы управления САУ БЛА. Результаты выполненных измерений и вычислений фиксируются внешним информационным накопителем.

На рис. 1 обозначено: БПО – блок предварительной обработки; БОПО – блок определения параметров ориентации; БО – блок определения; БСО – блок статистической обработки; ВИ – внешняя информация; НУИП – блок вычисления начальных условий и параметров; МНУСТ – модель нормального ускорения силы тяжести (УСТ); БРФА – блок реализации функциональных алгоритмов БГНК; БИ – блок измерений; ОФК – блок реализации алгоритма оптимального фильтра Калмана; БВДП – блок вычисления дополнительных параметров; БВМС – блок вычисления поправки на магнитное склонение $\Delta\psi_{mc}$; A_{OE} – матрица ориентации базовой правой Земной экваториальной системы координат (ЭСК) (начало в центре Земли, орты ξ , η расположены в плоскости экватора, где ξ – на линии пересечения плоскости экватора с Гринвичским меридианом, а орт ζ направлен по вектору $\mathbf{u}$ угловой скорости вращения Земли) относительно связанной с объектом системы координат (ССК); индексом $_0$ обозначены начальные условия соответствующих параметров; БП – блок пересчета показаний акселерометров в ЭСК; ψ , ϑ , γ – курс, тангаж, крен; ω – вектор абсолютной угловой скорости и $\mathbf{n}$ – вектор кажущегося ускорения объекта, измеренные БГ и БА; индексом $_o$ обозначены вектора, представленные в проекциях на оси ССК (без данного индекса – на оси ЭСК); $\mathbf{B}$ – вектор индукции геомагнитного

поля; $\mathbf{R}$ – геоцентрический радиус-вектор местоположения объекта; $\mathbf{U}$ – вектор относительной скорости объекта; индексы c и m отвечают параметрам, вычисляемым СНС и МКС, индекс A – алгоритму акселерометрического построителя вертикали; символ δ обозначает погрешность соответствующей величины, $\hat{}$ – ее оценку; $\omega_{Ш}$ – частота М. Шулера; $F_{N_{ГЦ}}$, F_R – проекции напряженности нормального поля тяготения Земли на оси сопровождающей геоцентрической системы координат (СК); $\mathbf{F}$ – матрица динамики; $\mathbf{G}$ – матрица шумов системы; $\mathbf{H}$ – матрица измерений; $\mathbf{Q}$ – матрица дисперсий шумов системы; $\mathbf{L}$ – матрица дисперсий шумов измерений; $\mathbf{P}$ – апостериорная матрица ковариации; $\mathbf{K}$ – матрица коэффициентов усиления ОФК; $\mathbf{z}$ – вектор измерений; $\mathbf{\Lambda}=(\lambda_0, \lambda_1, \lambda_2, \lambda_3)^T$ – вектор параметров Родрига-Гамильтона, характеризующий поворот ССК относительно инерциальной СК; $\mathbf{A}$ – вектор аномалии УСТ; $\mathbf{A}_n$ – $\mathbf{A}$, заранее известный с ограниченной степенью точности (при отсутствии исходных данных $\mathbf{A}_n=0$); $\boldsymbol{\mu}$ – вектор коэффициентов затухания корреляционных функций, выбранных для описания соответствующих проекций случайных составляющих аномалии УСТ, $\boldsymbol{\sigma}$ – вектор СКО проекций случайных составляющих аномалии УСТ; t – время; $\boldsymbol{\rho}=(\rho_0 \quad \dots \quad \rho_m)^T$, $\dot{\boldsymbol{\rho}}=(\dot{\rho}_0 \quad \dots \quad \dot{\rho}_m)^T$ – вектора псевдодальностей и псевдоскоростей $(m+1)$ используемых навигационных спутников; α , β – углы атаки и скольжения; $\mathbf{V}$ – вектор воздушной скорости; $\Delta\alpha_w$, $\Delta\beta_w$ и $\Delta\alpha$, $\Delta\beta$ – приращения углов и систематических составляющих скоростей вертикального и бокового ветра соответственно.

Программно-аппаратный вычислительный модуль (ВМ) БГНК включает в себя аппаратную платформу, функционирующую под управлением операционной системы реального времени (ОСРВ) и прикладное программное обеспечение (ПО), реализующие функциональные алгоритмы комплекса.

На рис. 2,а представлена принципиальная схема аппаратного обеспечения ВМ [2]. Анализ существующей элементной базы, применяемой в современных вычислительных комплексах авиационного назначения, показал, что рациональным вариантом для построения малогабаритного ВМ является выбор современного микроконтроллера (МК) в качестве вычислительного ядра. ВМ целесообразно выполнить в виде малогабаритной печатной платы с размещенными на ней функциональными элементами. В качестве основы для реализации аппаратной платформы выбран 32-разрядный МК AT91SAM9G20 [3] компании Atmel. МК выполнен по архитектуре ARM9 в 217-выводном корпусе и имеет тактовую частоту 400 МГц. В нем присутствует аппаратная поддержка работы с подключаемой памятью, а также поддержка реализованных на плате интерфейсов RS-485, RS-232, USB. ВМ помимо МК содержит блок питания с разъемом питания J1, МК DD1, флэш-память DC1 увеличенного объема, оперативную память DC2, выводы интерфейсов J2, J3, J4, J5, J6. Блок питания предназначен для формирования напряжений питания 5 В, 3,3 В и 1 В из подключаемого по разъему J1 питания

5 В. Флэш-память DC1 предназначена для хранения исполняемого кода программ и хранения результатов измерений и вычисленных параметров. Оперативная память DD2 необходима для временного хранения данных и команд, исполняемых процессорным ядром МК. МК DD1 предназначен для исполнения программы, приема и передачи данных с внешних интерфейсов и работы с памятью. Адресация памяти производится по шине «Адрес», обмен данными производится по шине «Данные». При этом малые габариты ВМ (88х99 мм) достигаются использованием технологии поверхностного монтажа компактно расположенных SMD-компонентов на плате и трехслойной структурой печатной платы толщиной в 1 мм. Слои металлизации и дополнительное заземленное металлизированное покрытие верхнего и нижнего слоев печатной платы позволяют повысить помехозащищенность устройства, а установка на плату только необходимых БГНК узлов, рациональное расположение элементов и их связей позволяет минимизировать массу и габариты модуля, его энергопотребление.

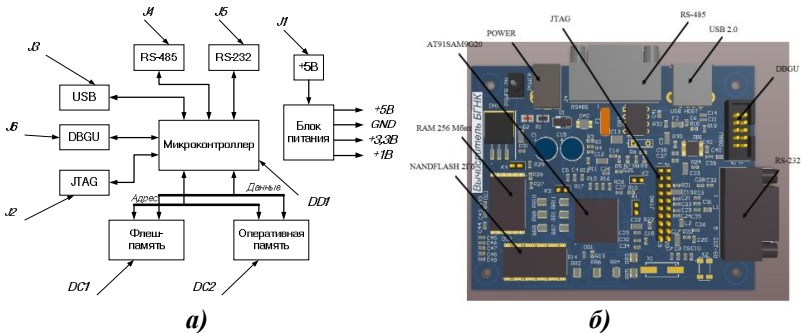


Рис. 2. Принципиальная схема ВМ (а) и его внешний вид (б)

ВМ обеспечен прикладным ПО, разработанным на языке С в среде разработки приложений для МК систем IAR Embedded Workbench. Прикладное ПО реализует функциональные алгоритмы БГНК, а также алгоритмы обмена информацией с измерителями, накопителем и системой индикации. Загрузка ПО в МК производится по интерфейсу JTAG, размещенному на плате модуля с помощью специализированных программных средств, таких как IAR Embedded Workbench. Помимо этого в состав ПО ВМ входит ОСРВ, под управлением которой функционирует прикладное ПО БГНК. Выбранный МК позволяет работать под управлением различных ОСРВ, таких как RT Linux, Windows CE и др. Основными задачами ОСРВ в ВМ являются управление процессами, динамическое распределение памяти, управление таймерами, обеспечение взаимодействия между процессами и их синхронизация, а также контроль устройств ввода/вывода.

На рис. 2,б показан внешний вид прототипа ВМ для полунатурных и экспериментальных исследований. ВМ БГНК предназначен для реализации его алгоритмов в режиме реального времени. Специальное ПО комплекса может быть реализовано как для запуска на МК без предустановленной операционной системы, так и для запуска в ее среде, в том числе и ОСРВ. Исходные данные для решения уравнений в рамках функциональных алгоритмов БГНК могут поступать как из файлов на сменном носителе, так и в реальном времени по одному или нескольким интерфейсов от подключенных измерителей. Результаты измерений и вычислений также могут быть записаны в память устройства либо переданы по одному из интерфейсов потребителю.

Таким образом, ВМ БГНК представляет собой компактную плату с весьма ограниченным энергопотреблением, которая может быть интегрирована совместно с другими модулями комплекса в состав бортового оборудования малоразмерного авиационного или морского носителя. Производительный микропроцессор в составе применяемого в модуле 32-разрядного МК работает с тактовой частотой 400 МГц и построен на базе современной архитектуры ARM, что обеспечивает высокую вычислительную производительность модуля. В состав ВМ входит микросхема ОЗУ объемом 64 Мб и ПЗУ объемом 2 Гб на основе микросхемы flash-памяти. Входящие в состав ВМ интерфейсы RS-485 и RS-232, подключенные к соответствующим приемопередатчикам МК и обладающие скоростью передачи данных 38 Кб/сек, могут качественно обеспечивать обмен информацией с измерительными устройствами, входящими в состав БГНК (например, трехосными БГ и БА, приемником СНС и др.). Температурный диапазон работы ВМ составляет от -40 до +85° С. Предполагается, что данный ВМ качественно и надежно обеспечит функционирование БГНК, по крайней мере, в его минимальном составе. Несмотря на свое целевое назначение, данный ВМ может быть использован не только в составе БГНК, но и также в составе других информационно-измерительных систем и комплексов, например, решающих задачи ориентации, навигации, сбора и оперативной обработки данных различных видов, а также управления малоразмерных подвижных объектов широкого класса и назначения.

1. Афонин А.А., Сулаков А.С., Ямашев Г.Г., Михайлин Д.А., Мирзоян Л.А., Курмаков Д.В. О возможности построения бесплатформенного управляющего навигационно-гравиметрического комплекса беспилотного летательного аппарата// Труды МАИ. – 2013. – №66.
2. Афонин А.А., Курмаков Д.В., Ямашев Г.Г., Сулаков А.С. Разработка малогабаритного микроконтроллерного вычислительного модуля для систем и комплексов ориентации, навигации, гравиметрии и управления аэроморских динамичных подвижных объектов// Труды МАИ. – 2013. – №66.
3. URL: <http://www.atmel.com/devices/sam9g20.aspx> (дата обращения: 01.11.2012).

ОПТИМИЗАЦИЯ АППАРАТНОГО РЕСУРСА ПРИКЛАДНОЙ ЗАДАЧИ ЗА СЧЕТ АРХИТЕКТУРНЫХ ОСОБЕННОСТЕЙ КРИСТАЛЛОВ ПЛИС*

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
simans2002@mail.ru*

В последнее время стремительно развиваются вычислительные системы с перестраиваемой (реконфигурируемой) архитектурой, предназначенные для решения вычислительно трудоемких задач.

Реконфигурируемые вычислительные системы (РВС), как правило, используют в качестве основного вычислительного ресурса программируемые логические интегральные схемы (ПЛИС), объединенные в единое вычислительное поле. Высокая реальная производительность РВС достигается за счет адаптации архитектуры самого вычислителя под структуру решаемой задачи.

Программирование РВС основано на парадигме структурной организации вычислений, согласно которой программа прикладной задачи формируется в виде информационного графа, вершины которого соответствуют операциям, а дуги – информационным зависимостям между ними [1] (в зарубежной литературе она известна как *dataflow programming paradigm* [2]). В отличие от процедурного программирования, когда текст исходной программы транслируется компилятором в набор инструкций, последовательно во времени исполняемых на процессоре (или совокупности процессоров), текст структурной программы для РВС преобразуется с помощью программы-синтезатора в совокупность логических блоков, команд функциональных устройств, расположенных в кристаллах ПЛИС, и связей между ними. Прикладная задача может быть решена на РВС только в том случае, если количество аппаратного ресурса, необходимого для реализации информационного графа, не превышает количество аппаратного ресурса вычислительного поля целевой РВС. Если же количество необходимого аппаратного ресурса задачи превышает количество доступного аппаратного ресурса ПЛИС целевой РВС, то необходимо использовать либо методы структурно-процедурной организации вычислений, либо методы сокращения необходимого аппаратного ресурса.

* Работа выполнена при финансовой поддержке Министерства образования и науки РФ, уникальный идентификатор проекта RFMEFI57814X0006.

Известные методы оптимизации информационного графа задачи (удаление тупиковых вершин, удаление висячих вершин, удаление дублирующих вершин, оптимизация элементов синхронизации и т.п.) [3] сокращают аппаратный ресурс за счет оптимизации (изменения) структуры информационного графа.

Другим подходом к сокращению используемого аппаратного ресурса является оптимизация, учитывающая архитектурные особенности кристалла ПЛИС.

Общая структура кристалла ПЛИС фирмы Xilinx представлена на рис. 1а. Блоки ввода/вывода располагаются по периферии кристалла, а конфигурируемые логические блоки (КЛБ), предназначенные для выполнения простых логических функций от нескольких переменных, а также функций триггера, расположены в виде матрицы в центре, между ними расположены конфигурируемые межсоединения.

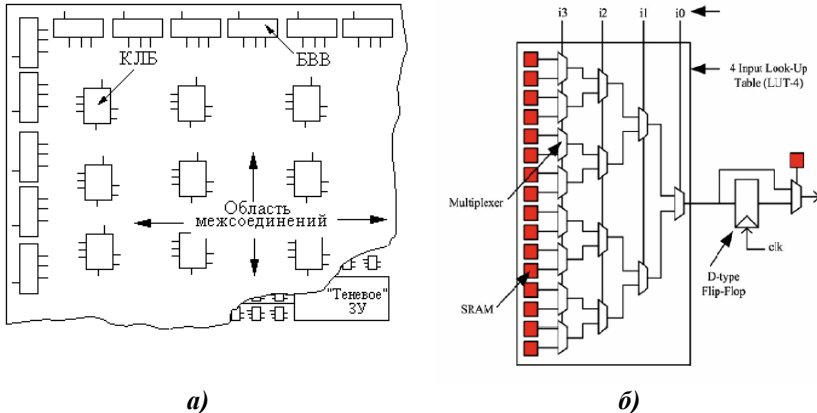


Рис. 1. Архитектурные особенности ПЛИС

а) упрощенная структура ПЛИС; б) пример четырехбитного логического элемента

Сложность и структура КЛБ определяются производителем, КЛБ может состоять из одного или нескольких базовых логических элементов (BLE – Basic Logic Element) или просто Logic Element (LE). В ПЛИС обычно используются так называемые LUT-based (LUT – Look-Up Table, таблица преобразования) BLE, которые позволяют реализовать логическую функцию от n переменных на основе таблицы истинности.

BLE содержат 2^n конфигурационных регистров для хранения таблицы истинности и 2 регистра для управления выходом. Пример BLE для ПЛИС Virtex-4 фирмы Xilinx ($n=4$) представлен на рис. 1б. Как пра-

вило, кристаллы ПЛИС фирмы Xilinx обладают удвоенным количеством триггеров по сравнению с количеством LUT элементов.

Разработанный метод оптимизации состоит в поиске и выделении клик логических элементов и их замене одним элементом BLE, что позволяет сократить используемый аппаратный ресурс. Под кликом понимается такая группа операционных вершин информационного графа, у которой количество входов не превышает заданное число и имеется всего один выход. Будем рассматривать клики, состоящие из операционных вершин, выполняющих логические операции (И, ИЛИ, НЕ и их производные).

На рис. 2а представлен пример клики (кружками обозначены операционные вершины, а стрелками – ребра графа), имеющей 4 входа. Непосредственная реализация данной клики на аппаратном ресурсе ПЛИС требует использовать 3 LUT элемента. Однако некоторые синтезаторы (например ISE фирмы Xilinx) способны автоматически находить подобные клики и отображать их на кристалл ПЛИС в виде таблицы истинности (рис. 2б), реализация которой требует использования одного LUT элемента. Единственным ограничением при реализации клики, состоящей из логических элементов, является число входов клики, которое не должно превышать количество битовых входов элемента LUT целевой ПЛИС. В противном случае «большая» клика делится на клики меньшего размера так, чтобы количество входов каждой клики не превышало количество битовых входов LUT элемента. При этом каждая "маленькая" клика реализуется на одном LUT элементе (рис. 3).

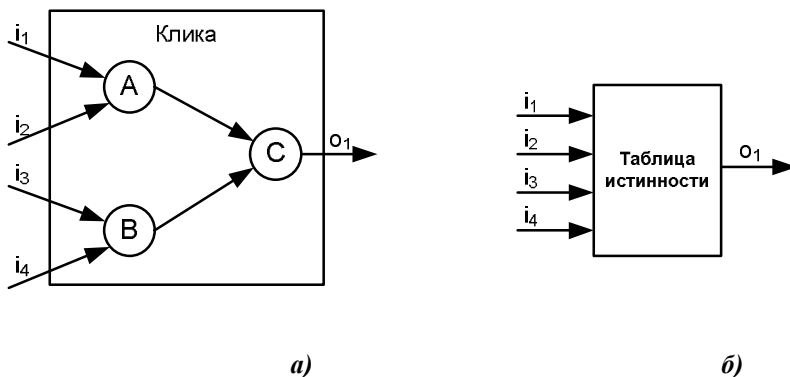


Рис. 2. Отображение клики на аппаратный ресурс ПЛИС

Усложним пример, приведенный на рис. 2, добавив в исходную клику операционную вершину, осуществляющую синхронизационную

задержку потока данных на 1 такт (рис. 4а, прямоугольником обозначена вершина, задерживающая поток данных).

Современные синтезаторы, в лучшем случае, разделяют данную клику на две клики меньшего размера (рис. 4б), что потребует задействовать два LUT элемента и один триггер для реализации данного подграфа на аппаратном ресурсе ПЛИС. Разработанный метод позволяет модифицировать исходный подграф, как показано на рис. 4в (вынести элементы задержки за пределы клики), для реализации которого потребуются два триггера и один LUT.

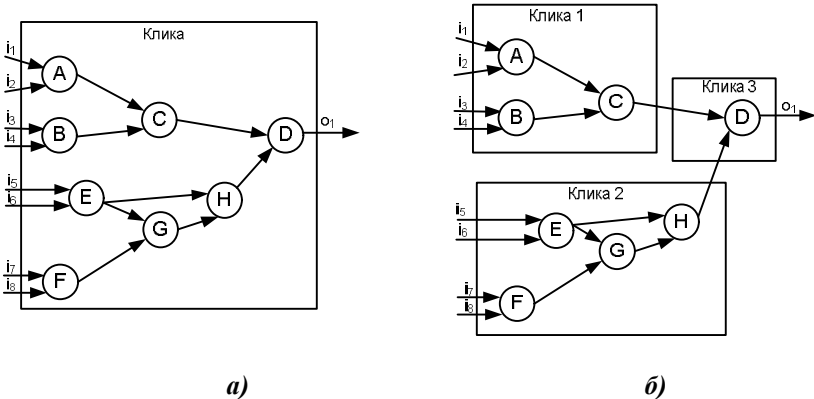


Рис. 3. Пример дробления «большой» клики

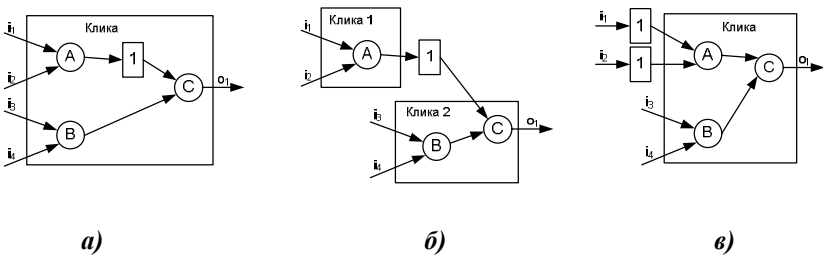


Рис. 4. Пример преобразования клики с триггером

В большинстве случаев выгоднее вместо LUT элемента использовать триггер. Для принятия верного решения (делать преобразование графа или не делать) необходимо точно знать процентное соотношение необходимого количества к общему количеству LUT элементов. Аналогичное соотношение необходимо и для триггеров. Если отношение LUT элементов больше отношения триггеров, то выгодны все преобразования информационного графа, ведущие к сокращению использования LUT элементов. Если же отношение LUT элементов меньше отношения

триггеров, то выгодны преобразования информационного графа, ведущие к сокращению использования триггеров.

Вывод. Автоматическое применение данного метода позволяет сбалансировать (уменьшить использование «критического» ресурса) использование триггеров и LUT элементов ПЛИС в решении прикладной задачи на PBC. Поскольку количество аппаратного ресурса оценивается по наименьшему доступному аппаратному ресурсу (критический ресурс), то можно сказать, что балансировка аппаратных ресурсов сокращает использование критического аппаратного ресурса целевой PBC. Применение данного метода позволяет сократить использование критического аппаратного ресурса на 3-5% в прикладных задачах в области линейной алгебры, математической физики, символьной обработки данных, цифровой обработки сигнала и других.

1. Котляров Д.В., Кутепов В.П., Осипов М.А. Граф-схемное потоковое параллельное программирование и его реализация на кластерных системах [Текст]. Теория и системы управления. М: Из-во РАН, 2005, №1.
2. Advances in dataflow programming languages. Wesley M. Johnston, J. R. Paul Hanna, Richard J. Millar. ACM Computing Surveys (CSUR) Volume 36 Issue 1, March 2004.
3. Левин И.И., Дордопуло А.И., Сорокин Д.А., Мельников А.К. Решение задач с существенно-переменной интенсивностью потоков данных на реконфигурируемых вычислительных системах // Вестник компьютерных и информационных технологий. – М.: Машиностроение, 2012. – №2. – С.49-56.

С.А. Бугаев, Е.Е. Завтур, Е.Ю. Шелестенко

АДАПТАЦИЯ АЛГОРИТМА ЦИФРОВОГО ФОРМИРОВАНИЯ КВАДРАТУРНЫХ СОСТАВЛЯЮЩИХ ДЛЯ РЕАЛИЗАЦИИ НА ПЛИС

*Научно-конструкторское бюро цифровой обработки сигналов
Южного федерального университета, г. Таганрог,
dsp@sfedu.ru*

Программируемые логические интегральные схемы (ПЛИС) становятся в последнее время все более распространенной и привычной элементной базой для разработчиков цифровых устройств.

Однако традиционные подходы, применяемые при реализации алгоритмов ЦОС на сигнальных процессорах, требуют определенной доработки при реализации их на ПЛИС.

Покажем это на примере реализации алгоритма цифрового формирования квадратурных составляющих (ЦФКС). Структурная схема алгоритма ЦФКС представлена на рисунке 1 [1].

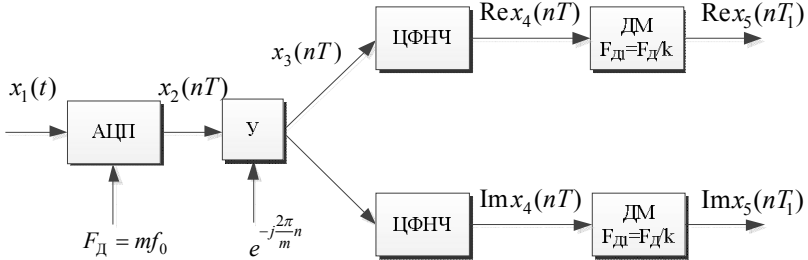


Рис. 1. Структурная схема алгоритма формирования квадратурных составляющих

Схема указанного алгоритма содержит аналого-цифровой преобразователь (АЦП), умножитель (У), два цифровых фильтра нижних частот (ЦФНЧ) с действительными коэффициентами и два дециматора (ДМ).

Функциональная схема модернизированного алгоритма для реализации на ПЛИС приведена на рис. 2.

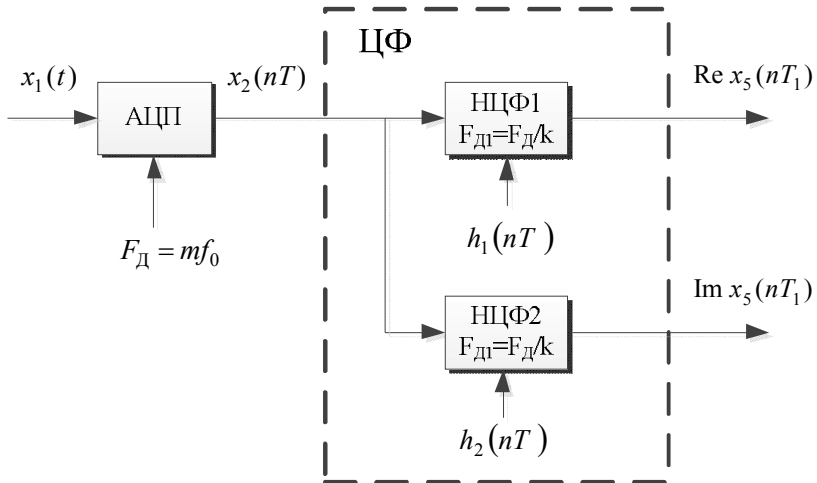


Рис. 2. Функциональная схема модернизированного алгоритма ЦФКС

Цифровой фильтр (ЦФ) объединяет в себе процедуры умножения отсчетов входного сигнала на комплексную экспоненту, цифровой фильтрации нижних частот и децимации с коэффициентом $K_d = k$.

Импульсные характеристики $h_1(nT)$ и $h_2(nT)$ нерекурсивных цифровых фильтров НЦФ1 и НЦФ2 соответственно равны

$$h_1(nT) = \operatorname{Re} \left(\tilde{h}^{\text{ЦФ}}(nT) \right) = h^{\text{ЦФНЧ}}(nT) \cdot \cos \left(\frac{2\pi}{m} (N_h - 1 - n) \right),$$

$$h_2(nT) = \operatorname{Im} \left(\tilde{h}^{\text{ЦФ}}(nT) \right) = h^{\text{ЦФНЧ}}(nT) \cdot \sin \left(\frac{2\pi}{m} (N_h - 1 - n) \right),$$

где $n = 0 \dots N_h - 1$.

Архитектурные особенности ПЛИС хорошо приспособлены для реализации таких операций как умножение, свертка и т.п. Это позволяет проектировать высокопроизводительные КИХ-фильтры высокого порядка с минимальными аппаратными затратами.

Рассмотрев основные алгоритмы цифровой обработки и пути их реализации, можно сделать следующие выводы.

Количество операций при реализации ЦФКС по схеме на рис.1 составляет $2L + 2N_h(L - N_h + 1)$ умножений и $2(L - N_h + 1) \cdot (N_h - 1)$ сложений, где L – длина входного массива данных.

Количество операций при реализации ЦФКС по схеме на рис.2 составляет $2N_h(L - N_h - 1)/k$ умножений и $2(L - N_h + 1) \cdot (N_h - 1)/k$ сложений.

Таким образом, выигрыш ресурсов ПЛИС после адаптации алгоритма ЦФКС составляет приблизительно k раз.

Это позволит в одной ПЛИС организовать многоканальную систему, причем каждый канал будет работать независимо в реальном масштабе времени.

В докладе представлены модифицированный алгоритм многоканальной первичной обработки сигналов локатора препятствий, материалы исследований и моделирования в среде MatLab для различных типов сигналов, а также результаты реализации алгоритма на ПЛИС фирмы Altera.

1. Цифровая обработка сигналов в системах и устройствах: монография /И.И. Маркович; Южный федеральный университет. – Ростов-н/Д: Издательство Южного федерального университета, 2012. – 236 с.

КОМПОНЕНТЫ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ РЕКОНФИГУРИРУЕМЫХ СУПЕРКОМПЬЮТЕРОВ НА ОСНОВЕ ПЛИС XILINX ULTRASCALE

*ФГУП «НИИ «Квант», г. Москва,
info@rdi-kvant.ru*

Введение

При создании реконфигурируемых суперкомпьютеров (РСК) используются современные программируемые логические интегральные схемы (ПЛИС). При этом элементная база ПЛИС за счет реконфигурации вычислительных ресурсов обеспечивает низкие мощности электропотребления, что является существенным вопросом эксплуатации центров обработки данных.

Перспективной областью применения РСК является использование ресурсов ПЛИС в облачных сервисах [3], [4]. Еще одна важная область применения, развиваемая в последнее время, – моделирование работы суперкомпьютера, объединяющего до нескольких миллионов процессорных ядер [5], [6].

Основными поставщиками рынка ПЛИС являются компании Xilinx и Altera. Для достижения возможности перепрограммирования ПЛИС используется технология на базе статической памяти. При построении РСК ФГУП “НИИ “Квант” применяет ПЛИС, развиваемые компанией Xilinx. В настоящее время Xilinx выпускает ПЛИС семейства Xilinx-7 и готовится к выпуску ПЛИС семейства Ultrascale [1]. Именно на эти типы ПЛИС ориентируются разработчики при построении РСК. Это вызвано в основном тем, что использование самой современной элементной базы позволяет улучшить технико-экономические параметры конечного изделия и успешно конкурировать на ряде приложений с решениями, использующими графические ускорители и универсальные микропроцессоры. Основными компонентами, на базе которых строятся реконфигурируемые суперкомпьютеры, являются реконфигурируемый вычислительный модуль (плата, содержащая ПЛИС) и реконфигурируемый вычислительный блок (набор плат с ПЛИС с соответствующей инфраструктурой электропитания, охлаждения и мониторинга). В докладе рассмотрены в сравнении компоненты РСК, выполненные на ПЛИС Xilinx-7 и Ultrascale.

Краткая характеристика нового семейства ПЛИС Xilinx Ultrascale

Новое семейство ПЛИС Xilinx Ultrascale обладает следующими отличительными особенностями:

- при производстве применяется новый техпроцесс – 20 нм, как следствие, удвоена логическая емкость по сравнению с 7 семейством;

- полностью пересмотрена система внутреннего тактирования и разводки, позволяющая приблизить новое семейство по классу к заказным микросхемам ASIC;
- увеличена производительность высокоскоростных вводов-выводов до 32.75 Gb/s (GTY);
- добавлен механизм непрерывного автоматического адаптивного выравнивания сигнала (для компенсации изменения температуры и напряжения питания);
- добавлен механизм контроля входного сигнала в виде двумерной глаз-диаграммы;
- во всех трансиверах реализована поддержка новейших стандартов PCI-E Gen3 (аппаратная поддержка) и Gen 4 (программная поддержка);
- аппаратно реализованы блоки 100G Ethernet, которые можно сконфигурировать как 10×10G или 4×25G;
- аппаратно реализованы Interlaken IP для высокоскоростного соединения между кристаллами, возможны конфигурации 12×12.5G или 6×25G.

Основные характеристики ПЛИС Xilinx Ultrascale в сравнении с ПЛИС Xilinx-7 приведены в таблице 1.

Характеристики ПЛИС Xilinx Ultrascale в сравнении с ПЛИС Xilinx-7

Таблица 1

Максимальное значение параметра	Kintex-7	Kintex-US	Virtex-7	Virtex-US
Логические ячейки, тыс.	478	1161	1995	4407
Блочная память, Мбит	34	76	68	115
Секций DSP	1920	5520	3600	2880
Приемопередатчики	32	64	96	104
Пиковая скорость передачи, Гбит/с	800	2086	2784	5101
Интерфейсы PCI Express (блоки)	1	4	4	6
Скорость обмена по интерфейсам памяти, Мбит/с	1866	2400	1866	2400

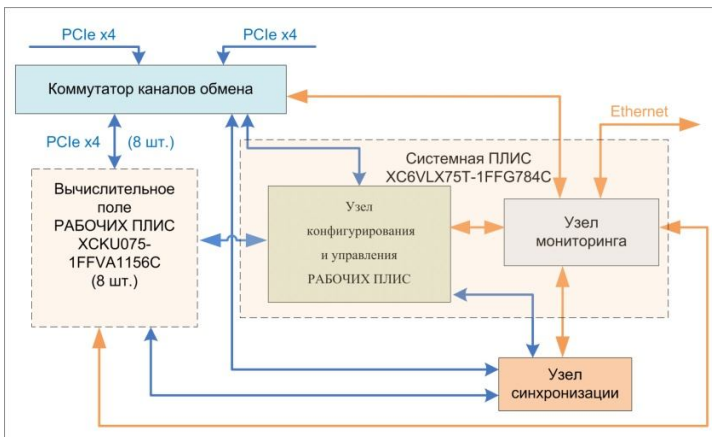
В настоящее время доступны инженерные образцы ПЛИС Kintex-Ultrascale-040, начало серийного выпуска этих микросхем намечено на 1 квартал 2015 года.

Компоненты РСК и их характеристики

Реконфигурируемый вычислительный модуль (PBM) на базе Xilinx Kintex-7 KB8-K7-410-01 «Топаз-2» разработан в ФГУП «НИИ «Квант». Основные технические характеристики модуля:

- размер модуля: 375×150 мм;
- сервисная ПЛИС: Xilinx Virtex-6 (LX75/130/195/240T) FF(G)784;
- сервисная ПЛИС поддерживает PCI Express Gen1 x4 или Gen2 x4;
- восемь рабочих ПЛИС: Xilinx Kintex-7 (XC7K325/410) FF(G)900;
- все рабочие ПЛИС поддерживают PCI Express Gen1 x4 или Gen2 x4;
- сеть PCI Express: одна СБИС PEX8648 коммутатора PCI Express Gen2 x4 (используются 11 портов), все ПЛИС подключены к PEX8648;
- микросхема FLASH памяти: одна конфигурационная Flash 512 Мбит;
- память SDRAM: 1Gb DDR3 SDRAM DDR-800;
- сеть Ethernet: Fast Ethernet Base-T.

Системная ПЛИС и 8 рабочих ПЛИС подключены к коммутатору PCI Express, реализованному на микросхеме PEX8648 (12 портов x4) фирмы PLX Technology. Объединение ПЛИС реализовано по схеме «звезда». Дополнительно все ПЛИС объединены в кольцо (кольцо Rocket IOx4), что позволяет организовать высокоскоростной канал обмена данными. Кроме этого, все ПЛИС подключены к общей шине для организации управления. Загрузку конфигурационных данных в рабочие ПЛИС осуществляет сервисная ПЛИС через 8 независимых каналов в режиме Slave Serial. Для организации мониторинга основных компонентов на плате реализован внешний сетевой интерфейс Ethernet, а для связи с внешними PCI Express устройствами – два канала PCIEx4. При проектировании PBM на базе ПЛИС Kintex Ultrascale KB8-KU-075-01 «Топаз-3» применены аналогичные решения. Структурная схема модуля приведена на рисунке.



Структурная схема модуля «Топаз-3»

На основе прогноза показателей стоимости и энергопотребления ПЛИС Ultrascale проведено сравнение технико-экономических параметров компонентов РСК, выполненных на ПЛИС Kintex-7-410 и Kintex-Ultrascale-040. Данные для предварительного расчета взяты из [2]. Результаты в относительных единицах представлены в таблице 2.

Сравнение технико-экономических параметров компонентов РСК, выполненных на ПЛИС Kintex-7-410 и Kintex-Ultrascale-040

Таблица 2

ПЛИС					
Стоимость		Количество ЛЯ		Энергопотребление	
Kintex-7	Kintex-US	Kintex-7	Kintex-US	Kintex-7	Kintex-US
1	0,5	1	1,13	1	0,5

Реконфигурируемый вычислительный модуль					
Стоимость		Количество ЛЯ		Энергопотребление	
Kintex-7	Kintex-US	Kintex-7	Kintex-US	Kintex-7	Kintex-US
1	0,7	1	1,13	1	0,55

Реконфигурируемый вычислительный блок					
Стоимость		Количество ЛЯ		Энергопотребление	
Kintex-7	Kintex-US	Kintex-7	Kintex-US	Kintex-7	Kintex-US
1	0,75	1	1,13	1	0,6

Технико-экономические параметры компонентов РСК будут уточнены после окончания разработки и изготовления опытных образцов реконфигурируемых блоков в 1 квартале 2015 года.

Заключение

В докладе рассмотрены в сравнении компоненты РСК, выполненные на ПЛИС Xilinx-7 и Ultrascale. Проведена предварительная оценка технико-экономических параметров компонентов РСК. Ожидается, что реконфигурируемый вычислительный блок на базе ПЛИС Kintex-Ultrascale-040 по сравнению с блоком на ПЛИС Kintex-7-410 будет на 25% дешевле, потребление электроэнергии снизится на 40%, количество логических ячеек увеличится на 13%. Выпуск таких блоков может начаться во 2 квартале 2015 года. Еще более лучшие характеристики можно ожидать после начала серийного производства в 2016 году ПЛИС Kintex-Ultrascale-075. Использование этого типа ПЛИС в реконфигурируемом блоке не потребует повторного проектирования платы вычислительного модуля.

Таким образом, переход на новую элементную базу ПЛИС позволяет существенно улучшить технико-экономические показатели компонентов РСК.

1. Xilinx Monthly Update. December 2013.
2. UltraScale architecture overview. FPGA platform marketing. December 2013.
3. Горбунов В.С., Лабошин Л.Ю., Лукашин А.А., Титов А.Г.. Исследование возможности предоставления реконфигурируемых вычислительных ресурсов на базе ПЛИС в виде облачного сервиса // Научный сервис в сети Интернет: все грани параллелизма: Труды Международной суперкомпьютерной конференции. (23-28 сентября 2013 г., г. Новороссийск). - М.: Изд-во МГУ, 2013. - 589 с.
4. Горбунов В., Соколов А. ПЛИС встраивается в облака // Суперкомпьютеры № 4 (16). 2013.
5. Елизаров С.Г., Лукьянченко Г.А., Корнеев В.В. Создание вычислительной системы для моделирования суперкомпьютера с производительностью эксафлопсного уровня. // «НСКФ-2012».
6. Елизаров Г.С., Горбунов В.С., Титов А.Г. Аппаратно-программная платформа моделирующей гетерогенной ВС (МГВС) // «НСКФ-2013».

Ю.Л. Зачесов, И.Н. Золотов

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ДОСТИЖЕНИЯ ЭКЗАФЛОПСНОЙ ПРОИЗВОДИТЕЛЬНОСТИ ДЛЯ ОДНОЙ ТРУДОЁМКОЙ ЗАДАЧИ

ФГУП «НИИ «Квант», г. Москва

В настоящее время выполняется работа по определению архитектурно-программных принципов и выбору элементно-конструктивной базы для создания эксафлопсных компьютеров. Большую роль в данной работе занимает моделирование для оценки производительности, а также имитационное моделирование и эмуляция и, в частности, создание эмуляторов элементов новых суперкомпьютеров на основе реконфигурируемых и графических процессоров.

В данной работе рассмотрено использование ПЛИС и графических процессоров в качестве эмуляторов ускорителей эксафлопсного компьютера. Для имитации задач, которые будут решаться на новом суперкомпьютере, выбрана известная трудоемкая задача – факторизация больших чисел.

Существование алгоритма, решающего задачу факторизации за полиномиальное время, заставило бы отказаться от RSA в будущем и скомпрометировало бы большое количество уже существующих систем. Однако самый быстрый алгоритм факторизации произвольных натуральных чисел, известный на сегодняшний день, имеет субэкспоненциальную оценку времени работы. Это значит, что он работает медленнее

полиномиального, но все-таки значительно быстрее экспоненциального. Благодаря этому приемлемый уровень безопасности в схеме шифрования RSA достигается при использовании ключей достаточно небольшого размера. Но прогресс в области компьютерных технологий и алгоритмов факторизации постоянно увеличивает нижнюю границу размера для ключа, который считался бы на текущий момент безопасным. Таким образом, исследование того, как изменится ситуация с появлением вычислительных машин экзотуровня, является актуальным.

Лучшим субэкспоненциальным алгоритмом просеивания является метод, который называется решето числового поля (number field sieve или NFS). Основная идея метода заключается в следующем.

Мы хотим разложить большое составное число n . Предположим, что у нас есть неприводимый многочлен $f(x)$ и число m , такое, что $f(m) \equiv 0 \pmod{n}$. Рассмотрим комплексный корень α многочлена $f(x)$ и кольцо $\mathbb{Z}[\alpha]$. $f(m) \equiv 0 \pmod{n}$ и $f(\alpha) = 0$, следовательно, есть естественный гомоморфизм колец $\varphi: \mathbb{Z}[\alpha] \rightarrow \mathbb{Z}_n$, который отображает α в m .

Теперь предположим, что у нас есть множество пар чисел (a, b) таких, что:

- произведение всех $(a - ab)$ – квадрат в кольце $\mathbb{Z}[\alpha]$, обозначим его γ^2 ;
- произведение всех $(a - mb)$ – квадрат в кольце $\mathbb{Z}$, обозначим его v^2 .

Заменим в выражении для γ корень α на m ; получим $\varphi(\gamma) \equiv u \pmod{n}$. Теперь $u^2 \equiv \varphi(\gamma)^2 = \varphi(\gamma^2) = \varphi(\prod (a - ab)) = \prod (\varphi(a - ab)) = \prod (a - mb) = v^2 \pmod{n}$, и мы, тем самым, сможем разложить n на множители.

Неприводимый многочлен f получается на первом этапе метода NFS. Значения a и b берутся из решета, о котором речь пойдет ниже. Чтобы $\prod (a - mb)$ было квадратом, нужно решить линейную систему на коэффициенты. Чтобы $\prod (a - ab)$ было квадратом, нужно решить линейную систему на коэффициенты в кольце $\mathbb{Z}[\alpha]$, если это хорошее кольцо с единственностью разложения. Теперь можно просто объединить две системы – нам нужно, чтобы оба свойства выполнялись.

Чем хорошо решето числового поля? Оценки трудоемкости основаны на X – количестве чисел, из которых можно сделать квадрат. У предыдущих методов $X = n^{1/2+\varepsilon}$, а в числовом решете получается $X = e^{c(\log n)^{2/3}(\log \log n)^{1/3}}$, что даёт общую оценку сложности $L_n \left[\frac{1}{3}, c \right] = e^{(c+o(1))(\log n)^{2/3}(\log \log n)^{1/3}}$. Теоретический рекорд: $c \approx 1,902$, обычно используют $c \approx 1,923$, но главное – основная асимптотика стала лучше.

Трудоемкость рекордных факторизаций по данным с сайта [1] показана в таблице.

Как известно, этап просеивания является наиболее трудоемким этапом метода NFS. На основе мультипликативного алгоритм просеивания был разработан алгоритм, позволяющий использовать ускорители различных типов для выполнения этого этапа метода NFS.

Таблица изменения трудоемкости

Год установки рекорда	Число битов в модуле факторизации	Трудоемкость алгоритма
1999	512	$2.6 \cdot 10^{17}$
2005	660	$5.4 \cdot 10^{18}$
2009	768	$1.4 \cdot 10^{20}$

При факторизации чисел, превышающих 768 двоичных разрядов, целесообразно использовать факторные базы с параметрами $B_A \geq 2^{36}$ и $B_R \geq 2^{32}$; для размещения элементов таких факторных баз требуется несколько Гбайт памяти. Для эмуляции будущего ускорителя экзамашины при решении задачи использовались современные ПЛИС и графические процессоры.

Блочная память одной ПЛИС не превышает нескольких Мбайт. Для размещения элементов таких факторных баз потребуется не меньше 1000 ПЛИС. Использование такого количества ПЛИС при обработке одной решетки будет являться неэффективным, так как время передачи данных между ПЛИС будет превышать время выполнения вычислений. С другой стороны, ПЛИС позволяет очень быстро обрабатывать факторные базы небольшого размера с параметрами $B_T = B_A \leq 2^{21}$.

Поэтому наиболее эффективным способом использования ПЛИС при факторизации больших чисел является декомпозиция задачи, при которой на ПЛИС обрабатываются элементы факторных баз с $p_i < 2^{21}$ (число таких элементов не превышает значение 3×10^5), а элементы факторных баз с большими значениями p_i (число таких элементов не превышает значение 2×10^{10}) обрабатываются на ускорителях других типов. Отметим, что при этом на ПЛИС будет выполняться не менее 85% вычислительной работы.

Была разработана программа, реализующая обработку таких элементов факторных баз на ПЛИС (Xilinx Kintex7-410T). Для этого в ПЛИС формируются не меньше $v=256$ обрабатывающих устройств (ОУ), каждое из которых включает в свой состав несколько сумматоров и умножителей, а также три блока памяти, в которых размещаются четырехразрядные элементы решетки и элементы из алгебраических и рациональных факторных баз (АФБ и РФБ).

В ПЛИС формируется распределительное устройство, основанное на нескольких v -разрядных циклических регистрах сдвига, обеспечи-

вающее на каждом t -ом, $t = \overline{1, v}$, такте работы устройства связь каждого j -ого, $j \in \{1, 2, \dots, v\}$, ОУ с s -ым ОУ, где $s \equiv j + t - 1 \pmod{v}$.

Алгоритм обработки решетки на ПЛИС состоит из трех шагов:

- загрузка элементов факторных баз в блочные памяти ПЛИС, или модификация этих элементов;
- вычисление адресов кандидатов;
- определение элементов факторных баз, на которые делятся многочлены, соответствующие кандидатам, передача выходных данных в управляющих процессорах (УП) для дальнейшей обработки.

Были получены оценки времени выполнения каждого из шагов алгоритма, и общего времени выполнения алгоритма на устройствах эмулирующих ускорители экзамашины, а также времени выполнения этих вычислений на одном ядре универсального процессора Intel Core2 Duo 2,16 ГГц. Получено, что можно достичь ускорения работы при выполнении просеивания для небольших элементов факторных баз, т.е. при выполнении более 85% задачи факторизации 768 битового числа — примерно в 450 раз по сравнению с одним универсальным процессором.

В графических процессорах для каждого кандидата определяются элементы факторных баз, являющиеся делителями соответствующего полинома. В УП передаются делители, соответствующие каждому из кандидатов. В УП осуществляется деление значений полиномов, соответствующих кандидатам, на переданные делители, и алгоритм кофакторизации.

Для минимизации количества обращений к факторным базам и достижения пикового быстродействия при работе с глобальной памятью применяется трехуровневое разбиение факторных баз — на гиперблоки, блоки и подблоки.

Использование ГП позволяет сократить время выполнения этапа просеивания при факторизации больших чисел примерно в 50 раз по сравнению программной реализацией алгоритма просеивания только на УП.

Совместное использование устройств, ориентированных на обработку небольших и больших элементов ФБ позволит ускорить время выполнения целевой задачи в 200 — 300 раз, что позволит выполнить этап просеивания при факторизации 1024 битового числа за 2 — 3 месяца на современных вычислительных системах. Переход к экзафлопсным архитектурам позволит ещё больше повысить производительность при решении рассмотренной задачи.

1. Информационное сообщение о рекордных факторизациях различных модулей RSA. <http://www.crypto-world.com/FactorRecords.html>

ОРГАНИЗАЦИЯ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ДЛЯ ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ТЕПЛОВИЗИОННЫХ СИГНАЛОВ

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
ovk@mvs.tsure.ru*

Спецификой многоэлементных фотоприемных устройств (МФПУ) инфракрасного диапазона является превышение (иногда значительное) фоновых и темновых составляющих сигналов над полезным сигналом. Другой особенностью является значительная разница между элементами МФПУ по чувствительности и темновому току, что создает структурный шум, который маскирует изображение объектов на малоконтрастной тепловой сцене [1]. Эти особенности потребовали разработки специализированных алгоритмов и аппаратно-программных средств вычислительных систем, предназначенных для предварительной обработки тепловизионных сигналов – сигналов МФПУ.

При этом тенденция увеличения количества фоточувствительных элементов МФПУ приводит к увеличению потока данных, поступающих с фотоприемника, что, в свою очередь, предъявляет требование повысить быстродействие вычислительной системы, предназначенной для предварительной обработки тепловизионных сигналов.

Под предварительной обработкой тепловизионных сигналов будем понимать:

- коррекцию неоднородности чувствительности элементов МФПУ;
- улучшение отношения сигнал/шум для сигналов МФПУ;
- улучшение контрастности тепловизионного изображения;
- формирование кадра тепловизионного изображения;
- исключение влияния на изображение дефектных элементов МФПУ.

Кроме этих основных функций система должна выполнять еще ряд вспомогательных функций, связанных с синхронизацией и управлением фотоприемным устройством, обеспечением взаимодействия с внешними устройствами (технологическими ПЭВМ и пультом, телевизионным монитором).

Таким образом, вычислительная система должна выполнять:

- формирование сигналов синхронизации и управления МФПУ;
- преобразование аналоговых сигналов в цифровые;
- формирование калибровочных корректирующих коэффициентов при изменении теплового фона для компенсации постоянной составляющей сигналов МФПУ;

- коррекцию неоднородности чувствительности элементов МФПУ;
- улучшение отношения сигнал/шум для сигналов МФПУ за счет накопления;
- деселекцию дефектных фоточувствительных элементов МФПУ;
- формирование кадра тепловизионного изображения;
- улучшение контрастности тепловизионного изображения;
- формирование видеосигнала тепловизионного изображения.

Для построения вычислительной системы, предназначенной для предварительной обработки тепловизионных сигналов, предлагается использовать программируемые логические интегральные схемы (ПЛИС). Архитектура ПЛИС обладает большой гибкостью, на ее основе можно создавать специализированные структуры обработки данных, позволяющие совмещать распараллеливание и конвейеризацию алгоритмов обработки пикселей теплового изображения.

Как правило, МФПУ имеют ограниченное количество сигнальных выводов: 1, 2, 4 или 8 [2, 3]. Следовательно, объединив в едином конструктиве восемь аналого-цифровых преобразователей и ПЛИС соответствующей сложности и быстродействия, можно получить универсальную вычислительную систему, предназначенную для предварительной обработки тепловизионных сигналов.

Структура подобной вычислительной системы, предназначенной для обработки сигналов МФПУ, приведена на рисунке. В ее состав входят:

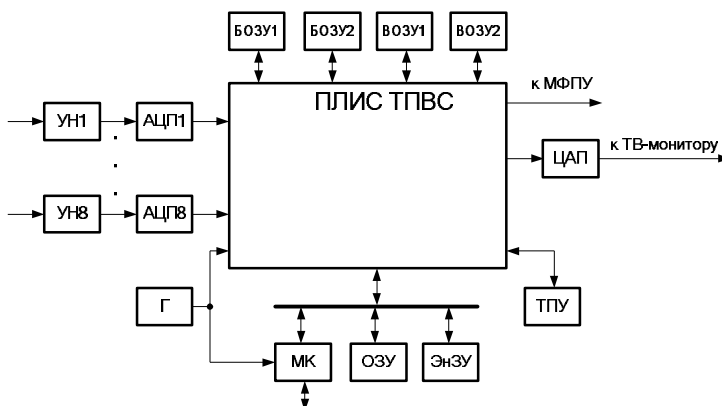
- усилители-нормализаторы аналоговых сигналов МФПУ (УН 1÷УН 8);
- аналого-цифровые преобразователи (АЦП 1÷АЦП 8);
- буферные запоминающие устройства (БОЗУ 1 и БОЗУ 2);
- видеопамять, состоящая из двух банков (ВОЗУ 1 и ВОЗУ 2);
- цифро-аналоговый преобразователь видеосигнала (ЦАП);
- программируемая логическая интегральная схема обработки тепловизионных сигналов (ПЛИС ТПВС);
- микроконтроллер (МК);
- оперативная память микроконтроллера (ОЗУ);
- энергонезависимая память микроконтроллера (ЭнЗУ);
- генератор синхросигналов (Г).

Такая структура вычислительной системы может быть использована при обработке сигналов различных типов МФПУ путем соответствующего программирования структуры ПЛИС ТПВС.

В ПЛИС ТПВС выполняются все основные функции, связанные с обработкой пикселей тепловизионного изображения, а микроконтроллер выполняет следующие функции:

- начальную (после включения питания) инициализацию системы;

- управление режимами функционирования системы;
- выполнение промежуточных вычислений при реализации последовательных алгоритмов;
- организацию взаимодействия с технологической ПЭВМ и технологическим пультом управления (ТПУ).



Структура подобной вычислительной системы, предназначенной для обработки сигналов МФПУ

Программное обеспечение вычислительной системы должно обеспечивать:

- инициализацию системы после подачи питающего напряжения;
- обработку калибровочных данных при выполнении калибровок по опорным сигналам (по "темному" и "светлому" фону) и по сигналам сцены;
- деселекцию дефектных фоточувствительных элементов МФПУ;
- изменение режимов работы системы от технологической ПЭВМ или от ТПУ;
- доступ к адресуемым регистрам, видеопамяти и буферной памяти системы.

Применение систем обработки тепловизионных сигналов на основе ПЛИС позволяет реализовать алгоритмы обработки пикселей тепловизионного изображения для различных МФПУ, используя оптимальное сочетание параллельных и конвейерных фрагментов алгоритмов.

1. Пономаренко В.П., Филачев А.М. Инфракрасная техника и электронная оптика. – М: Физматкнига, 2006, -336 с.
2. www.softdir.com
3. www.orion.ru

КОМПРОМИСС «ВРЕМЯ/ПАМЯТЬ» В РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

ЗАО «ИнформИнвестГрупп», г. Москва,
ak@iigroup.ru

Среди классов вычислительно трудоемких задач, для решения которых требуется разработка и создание высокопроизводительных вычислительных систем, можно выделить задачи, эффективно разбивающиеся на подзадачи, и временная сложность которых может быть значительно снижена путем сохранения уже вычисленных решений подзадач. Одним из наиболее экономически эффективных способов построения высокопроизводительных проблемно-ориентированных вычислительных систем (ПОВС) в настоящее время является использование реконфигурируемых мультиконвейерных вычислителей. Общая концепция создания проблемно-ориентированных реконфигурируемых мультиконвейерных вычислительных систем представлена в [1] и целом ряде работ тех же авторов.

В настоящей работе даются оценки необходимого объема памяти и ее производительности в зависимости от сложности решаемой задачи. Будем предполагать, что в момент t жизненного цикла ПОВС включает в себя $n = n(t)$ вычислительных элементов (ВЭ), каждый производительностью $p = p(t)$ элементарных подзадач в секунду, и от нее требуется решать проблемы сложностью $K = K(t)$ элементарных подзадач за время не более T . То есть мы предполагаем, что за время жизненного цикла в ПОВС для решения все более сложных задач может проходить как масштабирование путем увеличения количества ВЭ, так и модернизация путем увеличения производительности каждого отдельного ВЭ.

Предположим, что для решения проблемы со сложностью K используется такой метод разумного компромисса «время/память» (time-memory tradeoff) как метод так называемых радужных таблиц [2]. В [3] указывается, что опубликованные ранее анализы метода радужных таблиц [2], [4], [5], [6], [7] были проведены для модели с памятью с произвольным доступом (RAM model). В этой модели предполагается, что есть центральный процессорный модуль (CPU) и модуль памяти с произвольным доступом неограниченной емкости. В [3] оцениваются среднее время поиска и вероятность успеха в модели с внешней памятью (External Memory Model). Эта модель предполагает, что есть CPU и два уровня памяти: быстрая память ограниченной емкости и медленная память неограниченной ёмкости.

В отличие от предыдущих авторов, кроме двух перечисленных будем также рассматривать кластерную модель с распределенной памятью (Distributed Memory Model), в которой кроме n ВЭ также имеется k RAM-элементов объема m , а также одно медленное хранилище неограниченной емкостью.

Для того чтобы обеспечить время поиска не больше T при помощи заранее построенных таблиц, их объем M должен удовлетворять следующему требованию:

$$M \geq \max \left\{ \frac{lK}{\sqrt{2npT}}, \frac{lK}{ST} \right\}, \quad (1)$$

где l – объем памяти, необходимый для хранения одной «цепочки». Для широкого круга практических задач можно полагать $l = 16$ байт;

S – производительность памяти в операциях чтения с произвольным доступом в секунду (IOPS).

При этом

$$I = \frac{K}{nTM} \leq \sqrt{\frac{2pT}{n}}, \quad (2)$$

где $I = \frac{1}{n}S$ – удельная интенсивность запросов к памяти на 1 ВЭ, IOPS.

Таким образом, кроме суммарной производительности ВЭ критически значимыми параметрами являются суммарный объем памяти и скорость обращения к ней. Типовые значения IOPS при случайном чтении 4Кб блоков приведены в таблице по материалам из [8]. Использование RAID-контроллеров позволяет увеличивать общее количество IOPS до тех пор, пока определяющими не станут ограничения самого контроллера RAID и интерфейса доступа к хранилищу данных.

Значения IOPS для различных типов накопителей

	Устройство	IOPS
1	HDD 7200 RPMSATA	100
2	HDD 15KRPMSAS	180
3	SSD Intel X25 SATA II	5000 – 8000
4	SSD OCZ SATAIII	до 100000
5	RAMDISK, PCIe SSD	до 250000

Если оценка M в (1) не превосходит объем установленной оперативной памяти, то применима RAM модель построения ПОВС.

Если оценка M в (1) превосходит объем установленной оперативной памяти, а характеристики объема и производительности хранилища данных удовлетворяют (1), (2) то применима модель внешней памяти. При этом оказывается, что чем больше доступный объем внешней памя-

ти, тем меньше требования к ее производительности на рассматриваемом классе задач.

При дальнейшем росте сложности K решаемых задач наиболее адекватной становится кластерная модель с распределенной памятью. В зависимости от соотношений между M , $\frac{K}{M}$ и km в быстрой локальной памяти можно хранить либо сами радужные таблицы, либо только индексные файлы к ним. Тогда требования (2) будут относиться к производительности системы хранения индексных файлов, объем которой может быть существенно меньшим, чем M . Требования по производительности хранилища с самими таблицами объема M в этом случае будут зависеть от выбранного алгоритма индексирования и могут быть в результате существенно ниже, чем в (2).

Выводы

1. Оценки (1), (2) позволяют обосновать метод организации памяти и требования к ее характеристикам в зависимости от диапазона сложности задач, для решения которой предназначена ПОВС.

2. Если развитие ПОСВ планируется вести «экстенсивным» образом, когда относительный рост числа ВЭ опережает относительный рост производительности отдельных ВЭ, то при росте общей интенсивности запросов к памяти их удельная интенсивность снижается. При «интенсивном» развитии ПОВС, когда относительный рост производительности отдельных ВЭ опережает относительный рост числа ВЭ, растут и требования к удельной производительности системы памяти.

1. Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И. Реконфигурируемые мультиконвейерные вычислительные структуры. – Ростов н/Д: Издательство ЮНЦ РАН, 2008. – 320 с.
2. Philippe Oechslin. Making a Faster Cryptoanalytic Time-Memory Trade-Off - Proc. Advances in Cryptology (CRYPTO 2003), Santa Barbara, California, USA, Aug. 2003, pp. 617–630.
3. Jung Woo Kim, Jin Hong, Kunsoo Park. Analysis of the Rainbow Tradeoff Algorithm Used in Practice – JACR Cryptology ePrint Archive, 2013.
4. G. Avoine, P. Junod, and P. Oechslin, Characterization and improvement of time-memory trade-off based on perfect tables, ACM Trans. Inf. Syst. Secur., vol. 11, Jul. 2008.
5. J. Hong, The cost of false alarms in Hellman and rainbow tradeoffs, Des. Codes Cryptography, vol. 57, pp. 293–327, Dec. 2010.
6. J. Hong and S. Moon, A comparison of cryptanalytic tradeoff algorithms, J. Cryptology, vol. 26, pp. 559–637, Oct. 2013.
7. G. W. Lee, J. Hong, A comparison of perfect table cryptanalytic tradeoff algorithms, Cryptology ePrint Archive, Report 2012/540. <http://eprint.iacr.org/2012/540>
8. Как правильно мерять производительность диска – www.habrahabr.ru/post/154235/

ЗАЩИЩЕННАЯ КОМПЬЮТЕРНАЯ СЕТЬ РЕКОНФИГУРИРУЕМОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
korotkov@superevm.ru*

Современные сетевые угрозы представляют собой серьезную проблему для безопасности информационной инфраструктуры, бизнеса и защиты персональных данных. Для реконфигурируемых вычислительных систем (РВС), ресурсы которых доступны через глобальные или публичные компьютерные сети (Internet, WAN, MAN), эта проблема стоит особенно остро, т.к. простой дорогостоящего высокопроизводительного ресурса и несвоевременность получения результатов вычислений могут быть причиной крупных затрат.

В НИИ МВС ЮФУ в рамках ОКР «Разработка реконфигурируемой вычислительной системы РВС-7 и организация на ее основе производства реконфигурируемых вычислительных систем с производительностью до 10^{15} операций в секунду в одностоечном конструктиве 47U» была построена и принята в опытную эксплуатацию РВС-7. На РВС-7 могут решаться прикладные задачи, критичные к получению своевременного результата в обозначенных промежутках времени, такие как обработка метеоданных, управление химическими и биологическими процессами, цифровая обработка сигналов и изображений. Отказ системы вследствие несанкционированного доступа (НСД) и остановка вычислений может привести к серьезным последствиям, что недопустимо. Таким образом, необходима разработка способов защиты от НСД, сетевых атак и системы оповещения персонала.

Проведенный анализ информационной безопасности (ИБ) РВС-7, установленной в НИИ МВС ЮФУ и подключенной в сеть Internet, выявил основные виды сетевых угроз:

- доступ к ресурсам РВС по нешифрованному каналу связи;
- отсутствие персонифицированных учетных данных пользователей (в том числе на основе сертификатов);
- отсутствие системы предотвращения и обнаружения вторжений (IDS и IPS);
- использование стандартных tcp-портов;
- отсутствие оповещения администратора об обнаруженных аномалиях;
- внутренние угрозы.

Для устранения вышеуказанных недостатков предлагается организация виртуальной частной сети (VPN) на основе программного ком-

плекса OpenVPN, использование цифровых сертификатов пользователей, установка и конфигурирование системы обнаружения и предотвращения вторжений IDS/IPS Snort. Существующая топологическая схема сети PBC-7 показана на рис. 1.

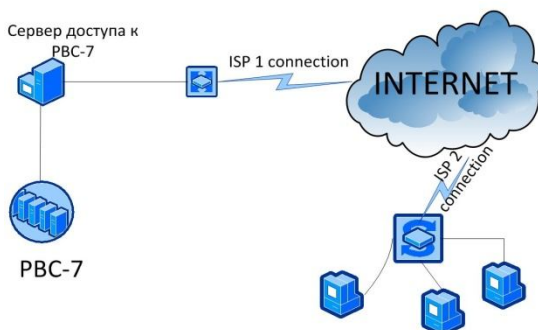


Рис. 1. Топологическая схема сети PBC-7

Обобщенная схема сети PBC представляет собой сегмент ЛВС, в котором расположены кластер суперЭВМ, управляющий сервер (УС) (шлюз во внешнюю сеть для кластера) и маршрутизатор, соединенный непосредственно с ISP (Internet Service Provider). Топология сети удаленных клиентских ПК в данном примере не имеет значения. ЛВС представляет собой сеть Ethernet топологии «иерархическая звезда» на типе кабеля UTP-5е. На УС запущена служба «Маршрутизация и удаленный доступ», предоставляющая прохождение сетевого трафика от PBC-7 к шлюзу в Интернет, таким образом, удаленные пользователи получают непосредственный доступ к PBC-7, обращаясь к статическому ip-адресу шлюза по ранее указанным протоколам.

При такой схеме взаимодействия пользователей с PBC через публичные сети очевидны недостатки:

- пользователи обращаются к PBC по статическому ip-адресу, используя стандартные средства удаленного доступа ОС Windows: протоколы RDP, SMB и RPC;
- сетевой трафик может быть перехвачен и относительно легко проанализирован, вследствие отсутствия СКЗИ (средства криптографической защиты информации);
- подверженность таким видам сетевых атак как DDoS, Ping flooding, «мусорные» IP-пакеты и шелл-код, инкапсулированный в IP.

Программный комплекс OpenVPN организует защищенное соединение во втором слое модели OSI «узел» - «узел» либо «узел» - «сеть» на основе персональных цифровых сертификатов, выданных пользователям, абстрагирует информационный поток от шлюза, сетей ISP и сетей магистральных провайдеров. Сертификаты обеспечивают аутентификацию пользователей в виртуаль-

ной сети. Настройка OpenVPN подразумевает смену стандартного порта 1194 на случайный непривилегированный порт из диапазона 1024–65536. Шифрование пакетов обеспечивается силами библиотеки OpenSSL, предоставляющей возможность работы с такими протоколами как AES и Blowfish.

Таким образом, поверх ip-сети на прикладном уровне поднимается виртуальная сеть между компьютером пользователя и непосредственно сервером доступа на его внутреннем интерфейсе. Следует принять во внимание, что уровень доверия к сети пользователя принят как изначально низкий. Таким образом, использование злоумышленниками специального программного обеспечения для перехвата и анализа трафика на участке сети от ЛВС-пользователя до так называемой «последней мили» положительных результатов не даст. Атака «Сканирование портов» также не даст полезной информации по причине отсутствия ответа «ECHO REPLY» от шлюза. При компрометации пользовательского сертификата задействуются возможности IDS Snort.

Система предотвращения вторжений (Intrusion Prevention System) Snort обеспечивает прослушивание дешифрованного сетевого трафика на внутреннем интерфейсе УС, анализ пакетов, эвристический поиск по содержанию, активное блокирование сетевых атак. В большинстве случаев IDS Snort защитит РВС от сетевых червей, атак с зараженного ПК, попыток определения ОС, поиска уязвимостей, запуска эксплойтов и передаст информацию администратору. Топологическая схема, обеспечивающая защиту РВС от всех популярных видов сетевых атак, показана на рис. 2.

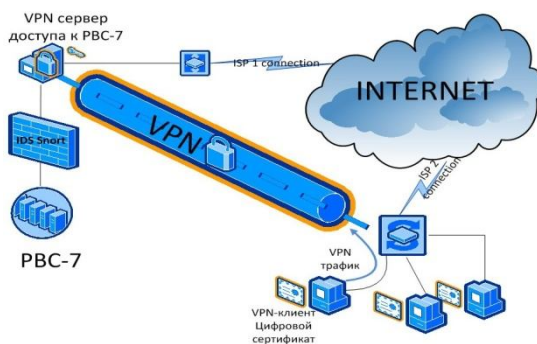


Рис. 2. Топологическая схема, обеспечивающая защиту РВС от всех популярных видов сетевых атак

Предложенная модель сетевой подсистемы и комплекса средств криптографической защиты информации (СКЗИ) обеспечит необходимый уровень защиты вычислительных ресурсов от НСД, своевременное оповещение администратора, персонализацию учетных данных и удобство использования. На скорость работы прикладных задач и результаты данное программное обеспечение не влияет.

И.И. Левин¹, А.И. Дордопуло², И.А. Каляев¹, Е.А. Семерников²

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ С РЕКОНФИГУРИРУЕМОЙ АРХИТЕКТУРОЙ НА ОСНОВЕ ПЛИС VIRTEX-7

*¹НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,*

*²Южный научный центр Российской академии наук, г. Ростов-на-Дону,
levin@mvs.tsure.ru, scorpio@mvs.tsure.ru, kaliaev@mvs.sfedu.ru,
semernikov@mvs.tsure.ru*

Введение

Реконфигурируемые вычислительные системы (РВС), содержащие в качестве основного вычислительного элемента множество кристаллов программируемых логических интегральных схем (ПЛИС), объединенных в единое вычислительное поле высокоскоростными каналами передачи данных, широко используются для решения вычислительно трудоемких задач в различных областях науки и техники, поскольку обладают рядом существенных преимуществ по сравнению с многопроцессорными вычислительными системами кластерной архитектуры: высокими реальной и удельной производительностями при решении задач, высокой энергоэффективностью и др.

Различные отечественные и зарубежные компании создают как отдельные ускорители с одним-двумя кристаллами ПЛИС, так и целые вычислительные комплексы. Такие фирмы как Nallatech [2] и Pico Computing [3] выпускают ряд ускорителей и несущих плат с небольшим числом (до четырёх) кристаллов ПЛИС, которые используются в создании серверов и гибридных кластерных систем фирмами HP и IBM. Компании Convey [4] и Maxeler Technologies [5] создают гибридные суперкомпьютеры на основе собственных гетерогенных кластерных узлов, каждый из которых может содержать от 1 до 4 кристаллов ПЛИС и несколько универсальных процессоров. Похожее решение используется и компанией SRC [6], которая выпускает узлы, названные MAP processor, для стойки (MAPstation) форм-фактором 1U, 2U и 4U. MAPstation 1U содержит один MAP processor. MAPstation 2U содержит до трёх MAP processor. MAPstation 4U может содержать до 10 различных модулей – MAP processor, модуль с универсальным микропроцессором или модуль памяти.

В отличие от перечисленных фирм в НИИ многопроцессорных вычислительных систем имени академика А.В. Каляева Южного федерального университета (НИИ МВС ЮФУ, г. Таганрог) разрабатываются суперкомпьютеры с возможностью объединения нескольких печатных плат в единый вычислительный контур, где основным вычислительным

ресурсом системы являются кристаллы ПЛИС. Методы разработки и создания таких систем успешно развиваются в НИИ МВС ЮФУ. Концепция построения РВС [7] позволила создать целый ряд высокопроизводительных систем различных архитектур и конфигураций, выпускаемых серийно и предназначенных для решения вычислительно трудоемких задач различных предметных областей. Опыт успешной эксплуатации в организациях и ведомствах Российской Федерации различных конфигураций ранее созданных РВС на основе ПЛИС Xilinx семейств Virtex-4, Virtex-5 и Virtex-6 использовался для разработки перспективных реконфигурируемых вычислительных систем на основе ПЛИС Xilinx Virtex-7, описание которых представлено в настоящей статье.

РВС на основе ПЛИС Xilinx Virtex-7

РВС на основе ВМ «Плеяда»

Реконфигурируемая вычислительная система РВС-7 на основе ПЛИС Virtex-7, разработанная по государственному контракту №14.527.12.0004 от 03.10.2011, содержит вычислительное поле из 576 микросхем ПЛИС Virtex-7 XC7V585T-FFG1761 объемом 58 миллионов эквивалентных вентилей каждая, конструктивно объединенных в один вычислительный шкаф высотой 47U с пиковой производительностью 10^{15} операций с фиксированной запятой в секунду.

Основным структурным компонентом РВС-7, предназначенным для установки в стандартную 19" вычислительную стойку, является вычислительный модуль (ВМ) 24V7-750 (ВМ «Плеяда»), в состав которого входят: четыре платы вычислительного модуля (ПВМ) 6V7-180, представленные на рис. 1; управляющий модуль УМ-7; подсистема питания; подсистема охлаждения и другие подсистемы. Фотография ВМ 24V7-750 представлена на рис. 2.



Рис. 1. Фотографии платы ПВМ 6V7-180: а) ПВМ №0-2 ВМ 24V7-750, б) ПВМ №3 ВМ 24V7-750 с оптическим разъемом для связи с другими ВМ 24V7-750



Рис. 2. Фотографии вычислительного модуля (ВМ) 24V7-750:
а) со снятой верхней крышкой, б) с установленной верхней крышкой

Производительность одной ПВМ 6V7-180 составляет 645,9 Гфлопс при обработке 32-разрядных данных с плавающей запятой, а производительность ВМ 24V7-750 составляет 2,58 Тфлопс при обработке 32-разрядных данных с плавающей запятой. Производительность РВС-7 при комплектации от 24 до 36 ВМ 24V7-750 составит от 62 до 93 Тфлопс при обработке 32-разрядных данных с плавающей запятой и 19,4 – 29,4 Тфлопс при обработке 64-разрядных данных с плавающей запятой. Областью применения РВС-7 и вычислительных комплексов на ее основе является решение задач цифровой обработки сигналов и многоканальная цифровая фильтрация.

РВС на основе ВМ «Тайгета»

На основе ПЛИС Virtex-7 также разработан новый вычислительный модуль «Тайгета» в конструктивном исполнении высотой 2U, предназначенный для установки в стандартную 19" вычислительную стойку. ВМ «Тайгета», представленный на рис. 3а, содержит четыре ПВМ 8V7-200, представленных на рис. 3б, соединенных быстрыми LVDS-каналами, встроенную управляющую ЭВМ, систему питания, систему управления, систему охлаждения и другие подсистемы.

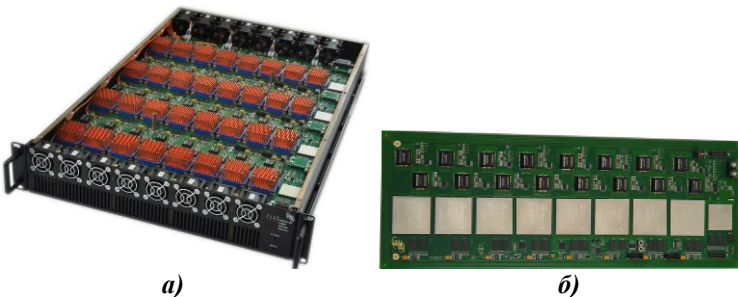


Рис. 3. Фотографии ВМ «Тайгета»: **а) фотография ВМ «Тайгета» со снятой верхней крышкой, б) фотография ПВМ 8V7-200**

ПВМ 8V7-200, лежащая в основе ВМ «Тайгета», представляет собой 20-слойную печатную плату с двухсторонним монтажом элементов, на которой располагаются 8 ПЛИС типа XC7VX485T-1FFG1761, содержащих 48,5 миллионов эквивалентных вентиляей, 16 микросхем распределенной памяти SDRAM типа DDR2 общим объемом 2 Гбайт, интерфейсы LVDS и Ethernet и другие компоненты.

Производительность одной ПВМ 8V7-200 составляет 667 ГФлопс при обработке 32-разрядных данных с плавающей запятой, а производительность ВМ «Тайгета» составляет 2,66 ТФлопс при обработке 32-разрядных данных с плавающей запятой. Производительность РВС на основе ВМ «Тайгета» при комплектации от 18 ВМ «Тайгета» составляет 48 Тфлопс при обработке 32-разрядных данных с плавающей запятой и 23 Тфлопс при обработке 64-разрядных данных с плавающей запятой. РВС на основе ВМ «Тайгета» позволяет сократить стоимость поставки вычислительной системы для задач определенного класса (например для задач символьной обработки), обеспечивая при этом такую же производительность, как и РВС-7 с 24 ВМ 24V7-750. Поэтому ВМ «Тайгета» являются наиболее предпочтительными для построения высокопроизводительных вычислительных комплексов для решения задач символьной обработки данных, поскольку обеспечивают существенное конкурентное преимущество по большинству технико-экономических параметров: удельной производительности, энергоэффективности и др.

Реконфигурируемый вычислительный блок «Калеано»

На основе ПЛИС Xilinx Kintex-7 также разработан новый реконфигурируемый вычислительный блок (РВБ) «Калеано» в конструктивном исполнении высотой 1U, предназначенный для обработки поступающих по Gigabit Ethernet каналу данных без поддержки IP-протоколов. Возможные области применения: символьная обработка, математическая физика, моделирование и вычислительный эксперимент, цифровая обработка сигналов, линейная алгебра и т.п.

Фотографии РВБ «Калеано» и печатной платы, содержащей на одной плате как микросхемы ПЛИС, так и модуль загрузки и управления, представлены на рис. 4.

РВБ «Калеано» содержит шесть ПЛИС Kintex XC7K160T (16 миллионов эквивалентных вентиляей), встроенную управляющую ЭВМ, систему питания, систему управления, систему охлаждения и другие подсистемы. Все ПЛИС РВБ соединены между собой по принципу близкого действия с помощью LVDS-каналов, ко всем ПЛИС РВБ подключены модули динамической памяти емкостью 256 Мбайт каждый.

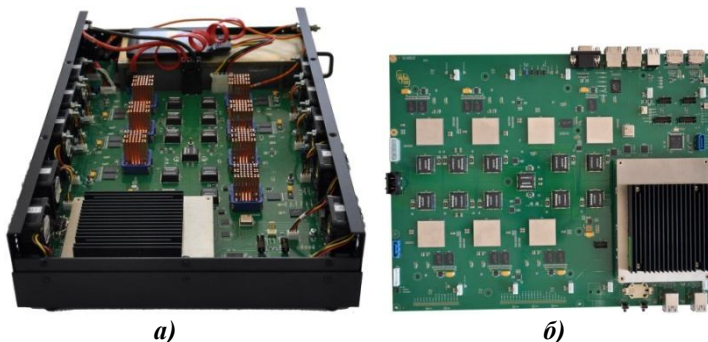


Рис. 4. Фотографии РВБ «Калеано»: а) со снятой верхней крышкой, б) фотография платы

Для управления и конфигурирования вычислительного поля РВБ используется процессорный модуль семейства COM-Express фирмы Kontron, установленный на плате вычислительного модуля. С его помощью осуществляются подготовка и отладка параллельной программы для решения вычислительно трудоемких задач на РВБ, формируются файлы исходных данных, которые вместе с исполняемым модулем задачи через шину PCI-Express загружаются в вычислительное поле по LVDS-каналу. После выполнения задачи результаты решения пересылаются в процессорный модуль COM-Express.

Пиковая производительность РВБ «Калеано» при выполнении операций обработки 32-разрядных данных с плавающей запятой на частоте 330 МГц составляет 150 Гфлопс и 75 Гфлопс при обработке 64-разрядных данных с плавающей запятой.

Программное обеспечение РВС на основе ПЛИС Xilinx Virtex-7

Для вычислительных систем на основе ПЛИС Virtex-7 сохраняется преимущество принципов программирования [7-10]: программирование всех рассмотренных вычислительных модулей осуществляется с помощью единого комплекса системного программного обеспечения, поддерживающего структурно-процедурные методы организации вычислений и определяющего не только организацию параллельных процессов и потоков данных, но и структуру вычислительной системы в поле логических ячеек ПЛИС. Наиболее характерной отличительной особенностью комплекса программного обеспечения РВС на основе ПЛИС Xilinx Virtex-7 является поддержка проблемно-ориентированных софт-архитектур, позволяющих создавать и программировать макро-объекты, представляющие собой совокупность вычислительных устройств, выполняющих определенную группу команд и соединенных

между собой коммутационной системой. Это обеспечивает при тех же принципах программирования и использовании языка высокого уровня для программирования РВС возможность простой адаптации программных компонентов средств разработки для РВС при переходе на новые топологии ПБМ без внесения существенных изменений в код программных компонентов комплекса, а также позволяет сократить время решения прикладных задач.

Большинство существующих коммерческих систем проектирования (Xilinx ISE, Altium Designer и другие) обеспечивает в рамках одного проекта работу только с одним кристаллом ПЛИС. Поэтому при разработке конфигурации для нескольких ПЛИС инженеру-схемотехнику приходится самому распределять элементы вычислительной структуры алгоритма решаемой задачи между различными проектами, которые будут соответствовать определённым кристаллам ПЛИС многокристальной РВС и учитывать топологию связей между кристаллами ПЛИС. Учет особенностей внутренней архитектуры, топологии и элементной базы РВС существенно усложняет специалисту-схемотехнику разработку конфигураций вычислительной системы прикладной задачи для многокристальных РВС и практически исключает возможность переноса (портации) готового решения на РВС другой конфигурации или архитектуры. Программирование РВС осуществляется в два этапа: на первом этапе схемотехник создает вычислительную структуру для решения прикладной задачи, а на втором этапе прикладной программист создает параллельную программу, определяющую потоки данных в созданной схемотехником вычислительной структуре. Поэтому сроки разработки решения прикладной задачи для РВС достаточно велики и составляют 4-9 месяцев.

Для программирования РВС, созданных в НИИ многопроцессорных вычислительных систем Южного федерального университета (г. Таганрог), используется разработанный комплекс программного обеспечения [1, 7, 9, 11], поддерживающий структурно-процедурные методы организации вычислений и определяющий как структуру вычислительной системы в поле логических ячеек ПЛИС, так и организацию параллельных процессов и потоков данных. Для РВС-7 преемственность принципов программирования [7-10] сохранена: программирование РВС-7 осуществляется на языке высокого уровня COLAMO [1, 11], в результате трансляции которого автоматически формируется конфигурация вычислительной системы в виде файлов конфигурации кристаллов ПЛИС (структурная составляющая параллельной программы) и параллельная программа, управляющая потоками данных и организацией вычислительного процесса в РВС. Отличительными особенностями комплекса программного обеспечения на основе языка про-

граммирования COLAMO по сравнению с известными средствами разработки MittrionC [12] и CatapultC [13], являются автоматическое размещение, синхронизация и создание конфигурации для многокристальных PBC, высокий процент заполнения кристалла (60-90%) и высокие частоты работы (250 – 350 МГц).

Средства программирования прикладных задач на языке высокого уровня COLAMO содержат следующие основные компоненты:

- транслятор языка программирования COLAMO, осуществляющий трансляцию исходного кода на COLAMO в информационный граф параллельной прикладной программы;
- синтезатор масштабируемых схемотехнических решений на уровне логических ячеек ПЛИС-7 Fire!Constructor, осуществляющий отображение полученного от транслятора языка программирования COLAMO информационного графа на архитектуру PBC, размещение отображенного решения по кристаллам ПЛИС и автоматическую синхронизацию фрагментов информационного графа в разных кристаллах ПЛИС;
- библиотеку IP-ядер, соответствующих операторам языка COLAMO (функционально-законченных структурно-реализованных аппаратных устройств), для различных предметных областей и интерфейсов для согласования скорости обработки информации и связи в единую вычислительную структуру.

Заключение

Для разработанных ВМ на основе ПЛИС Xilinx Virtex-7 было проведено исследование реальной производительности созданных аппаратных средств. В качестве задачи для исследования производительности ВМ 24V7-750 и ВМ «Тайгета» использовалась задача многоканальной цифровой фильтрации для мониторинга систем цифровой связи третьего и четвёртого поколений (КИХ-фильтр высокого порядка), оперирующая данными с фиксированной запятой различной разрядности (16, 24, 32, 34 разряда). Поэтому полученные значения реальной производительности представлены в оп/с, а не в операциях с плавающей запятой: реальная производительность ВМ 24V7-750 составляет 25 Топ/с, реальная производительность ВМ «Тайгета» – 27 Топ/с, а реальная производительность РВБ «Калеано» – 5,6 Топ/с. Это позволяет достигать реальной производительности $0,9 \cdot 10^{15}$ оп/с в одной вычислительной стойке для PBC-7 и $0,486 \cdot 10^{15}$ оп/с для PBC на основе ВМ «Тайгета» при решении прикладных задач цифровой обработки сигналов.

Таким образом, конструктивные решения, положенные в основу перспективных вычислительных модулей на основе ПЛИС Xilinx Virtex-7 и Kintex-7, позволяют сосредоточить в пределах одной вычис-

лительной стойки высотой 47U мощный вычислительный ресурс и обеспечивают удельную производительность PBC на основе ПЛИС Xilinx Virtex-7 на уровне лучших мировых показателей для суперЭВМ с кластерной архитектурой. Это позволяет рассматривать PBC на основе ПЛИС Xilinx Virtex-7 как основу для создания высокопроизводительных вычислительных комплексов нового поколения, обеспечивающих высокую эффективность вычислений и близкий к линейному рост производительности при наращивании вычислительного ресурса.

1. Каляев А.В., Левин И.И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. М.: Янус-К, 2003. 380 с.
2. <http://www.nallatech.com/>
3. <http://picocomputing.com/>
4. <http://www.conveycomputer.com>
5. <http://www.maxeler.com/>
6. <http://www.srccomp.com/>
7. Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И. Реконфигурируемые мультиконвейерные вычислительные структуры /Изд. 2-е, перераб. и доп. / Под общ. ред. И.А. Каляева. Ростов-на-Дону: Изд-во ЮНЦ РАН, 2009. 344 с.
8. Каляев И.А., Левин И.И. Семейство реконфигурируемых вычислительных системы с высокой реальной производительностью // Труды международной научной конференции «Параллельные вычислительные технологии» (ПАВТ'2009). Нижний Новгород: электронное издание НГУ имени Н.И. Лобачевского, 2009. С.186-196.
9. Дордопуло А.И., Каляев И.А., Левин И.И., Семерников Е.А. Высокопроизводительные реконфигурируемые вычислительные системы нового поколения // Труды Международной суперкомпьютерной конференции с элементами научной школы для молодежи «Научный сервис в сети Интернет: экзафлопсное будущее». М.: Изд-во МГУ, 2011. С. 42-49.
10. Каляев И.А., Левин И.И., Семерников Е.А., Дордопуло А.И. Реконфигурируемые вычислительные системы на основе ПЛИС семейства Virtex-6 // Сборник трудов Международной научной конференции «Параллельные вычислительные технологии 2011» (ПАВТ 2011). Челябинск-М.: Издательский центр ЮУрГУ [Электронный ресурс], 2011. С. 203–210.
11. I.A. Kalyaev, I.I. Levin, E.A. Semernikov, V.I. Shmoilov. (2012). Reconfigurable multipipeline computing structures. Nova Science Publishers, New York.

СПЕЦИАЛИЗАЦИЯ ВЫЧИСЛЕНИЙ И ФУНКЦИОНАЛЬНО-ОРИЕНТИРОВАННЫЕ ПРОЦЕССОРЫ ДЛЯ ВСТРОЕННЫХ СУПЕРКОМПЬЮТЕРОВ*

*НПО автоматики имени академика Н.А. Семихатова,
Институт радиоэлектроники и информационных технологий
Уральского федерального университета,
Институт машиноведения Уральского отделения РАН,
г. Екатеринбург,
nicklookin@mail.ru, nicklookin1952@gmail.com*

Специализированный параллелизм и встроенные суперкомпьютеры

Главной тенденцией развития процессорных архитектур является переход к параллельным вычислениям. В то же время для многих известных архитектур суперкомпьютеров характерно замедление пропорционального роста производительности системы при наращивании числа процессоров. Это следствие *универсального параллелизма*, т.е. использования в качестве вычислительных модулей универсальных процессоров [1], поэтому большинство архитектур современных суперкомпьютеров являются плохо масштабируемыми.

В то же время существуют подходы к параллельным вычислениям, основанные на реализации *специализированного параллелизма* [1], когда архитектура системы ориентирована на максимально быстрое решение задач из определенной предметной области [2], [3].

Среди возможных областей применения специализированного параллелизма выделим класс суперкомпьютеров, встроенных в состав систем реального времени (EScomp [Embedded Supercomputers]) [4]. Главное требование к таким суперкомпьютерам – сверхвысокая производительность, при минимальных объемно-массовых характеристиках и потребляемой мощности. В пределе EScomp могут занимать ресурсы не более чем одной СБИС [5], что в ряде применений требуется уже сегодня.

Функционально-ориентированные процессоры

В настоящее время разработаны подходы к реализации высокопроизводительных вычислений, использующие специфику функций, процедур или алгоритмов. Одними из наиболее интересных являются

* Работа выполнена частично за счет программы Президиума РАН № 18 "Алгоритмы и математическое обеспечение для вычислительных систем сверхвысокой производительности" при поддержке УрО РАН (проект 12-П-1-1028)

результаты, полученные в течение ряда лет в Санкт-Петербургской научной школе СПбГТУ (ЛЭТИ), созданной на кафедре вычислительной техники профессором В.Б. Смоловым и развитой его коллегами и учениками. В 1970-1980-ых годах ими была сформирована и развита концепция *функционально-ориентированных процессоров* (ФОП) [6], которая, по мнению автора настоящей работы, наиболее верно отражает сущность эффективной реализации алгоритмов на специализированных процессорных архитектурах.

Разработанные в 1980-х годах методы и практические рекомендации в части обработки данных с помощью ФОП сегодня необходимо развить применительно к специфике построения вычислительных устройств в виде СБИС или их подсистем (кремниевых кластеров). По отношению к методам специализированного параллелизма это означает оптимизацию вычислений по критериям сложности. В качестве меры сложности в настоящей работе принимаем сложность функций в базисе схем из функциональных элементов L [7] как случай битовой сложности вычислений, введенной А.Н. Колмогоровым [8]. Соответственно, под быстрыми алгоритмами будем понимать алгоритмы, битовая сложность которых может быть оценена как $O(M(n) \log^c n)$, где c – некоторая константа [9].

В Группе микропроцессорных архитектур Института машиноведения УрО РАН в течение многих лет развивается теория проектирования ФОП для систем реального времени. Совместно с Институтом радиоэлектроники и информационных систем Уральского федерального университета (ИРИТ-РТФ УрФУ) и НПО автоматики имени академика Н.А. Семихатова разрабатываются реальные ФОП для конкретных систем.

В настоящей работе кратко рассматриваются результаты некоторых исследований и разработок в области ФОП для высокопроизводительных вычислительных средств систем реального времени.

Таблично-алгоритмические ФОП для вычисления математических функций

Математические функции – это широко используемый математический аппарат в различных алгоритмах систем реального времени. Существуют разнообразные методы нахождения значения функций [10]. Анализ показывает, что определенным преимуществом обладают методы полиномиальной сплайн-аппроксимации.

Базовой зависимостью, реализуемой в таблично-полиномиальных ФОП, является следующая:

$$P(X1, X2) = \sum_{i=0}^m A_{i(X1)} (X2)^i, \quad (1)$$

где $A_{i(X1)}$ – коэффициенты полинома, в нашем случае представляющие собой функции действительного аргумента m , для которых справедливо условие: $\lim_{i \rightarrow \infty} |A_{i(X1)}| \leq \alpha$, где α – произвольное действительное число; m – степень полинома. Полином в виде (1) обобщает множество полиномов различных типов.

Вычисление $P(X1, X2)$ по формуле (1) может быть произведено различными способами, каждый из которых обладает соответствующей степенью параллелизма

Базовая зависимость (1) с учетом описанного выше представления аргументов может быть реализована с помощью архитектуры ФОП, приведенной на рис. 1.

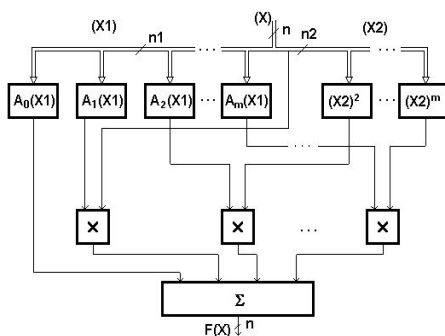


Рис. 1. Базовая архитектура таблично-алгоритмического ФОП

Основной принцип вычисления функций с помощью этого ФОП состоит в том, что код аргумента X разбивается на две независимые части – $X1$ и $X2$, а затем производятся независимые вычисления функций $A_{i(X1)}$, степеней $(X2)^i$ и перемножение их между собой с последующим итоговым суммированием.

Исследования показывают, что существенными параметрами для оптимизации и проектирования математических ФОП являются: минимально-допустимая степень полинома m для заданной погрешности вычисления функции; число опорных значений функции; аппаратная сложность вычисления элементарных логических функций.

Минимально-допустимая степень полинома может быть определена с помощью следующего рекуррентного алгоритма быстрого поиска:

$$m_{i+1} = \left\lceil \left[\sqrt{(n1 + \log_2 m_i - 1)^2 + n \cdot 2^{-(\log_2 m_i - 1)}} - (n1 + \log_2 m_i - 1) \right] m_i \right\rceil, \quad (2)$$

где $n1$ – "старшая" часть кода аргумента. Анализ алгоритма показал, что он сходится при любых начальных условиях, а предел сходимости является решением уравнения $\frac{n}{m} - \log_2 m = n1 - 0,5$. В работе [11] была получена верхняя оценка сложности вычисления полинома (1) в базисе схем из функциональных элементов. Она имеет вид

$$\frac{L_h^{FOP}}{m} = [n2^{n1} + (\sqrt{2}n - n1)^2], \quad (3)$$

где L_h^{FOP} – аппаратная сложность вычисления функции в базисе схем из функциональных элементов в классе архитектур таблично-алгоритмических ФОП.

Кроме того, было показано, что всегда найдется такое единственное разбиение аргумента X на "старшую" и "младшую" части $X1$ и $X2$, при котором достигается минимум сложности вычислений. Этот факт служит основанием для оптимизации СБИС ФОП по критерию аппаратных затрат.

В целом применение математических ФОП, оптимизированных по критериям аппаратных затрат или времени вычисления, может ускорить вычисление функций более чем на порядок относительно микропрограммной реализации, принятой в современных процессорных ядрах.

На основе предложенных подходов в течение последних десяти лет были разработаны ряд архитектур математических ФОП для одного из образцов бортовой ЦВМ, а также логический проект на IP-ядро таблично-алгоритмического ФОП для класса 64-разрядных вычислений в формате фиксированной точки.

Параллельные ФОП с архитектурой SIMD

ФОП данного вида предназначены для реализации алгоритмов широкого круга задач с явно выраженным параллелизмом – сортировка, поиск, классификация, линейная алгебра и т.п. В течение ряда лет нами разрабатывается подход к синтезу архитектур ФОП, основанный на ДН-преобразованиях ярусно-параллельных форм (ЯПФ) алгоритмических графов [8], где D – ширина алгоритмического графа, H – его высота [14]. В качестве примера можно привести результаты работ по созданию ФОП с параллельной архитектурой для реализации алгоритмов привязки в задаче корреляционно-экстремальной навигации.

Применение ДН-преобразований алгоритмических графов, в частности порождение сложных функций [15], позволяет строить их максимальные ЯПФ. В качестве примера на рис. 2 приведен граф алгоритма привязки с максимальной степенью параллелизма.

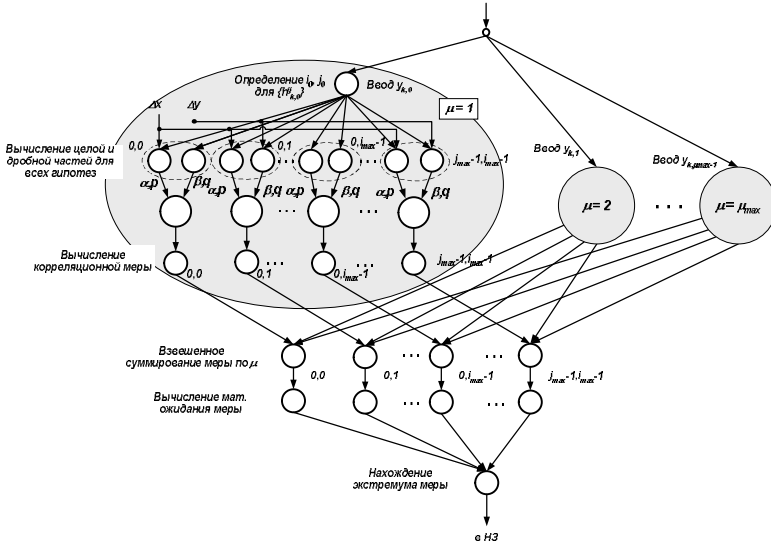


Рис. 2. Максимальная ЯПФ графа алгоритма привязки

Следующим этапом синтеза архитектуры ФОП является взвешивание вершин (т.е. функциональных преобразований алгоритма) максимальной ЯПФ оценками сложности вычислений в базисе схем из функциональных элементов СБИС.

Верхние оценки аппаратной и временной сложности вычислений для алгоритма привязки имеют вид [13]

$$\begin{cases} L_h^{(1)} < 60r^2 i_{\max} j_{\max} \mu_{\max}, L_t^{(1)} < 4r(14 + \lceil \log_2 \mu_{\max} \rceil), \\ L_h^{(2)} < 10r i_{\max} j_{\max}, L_t^{(2)} < 4r \lceil \log_2 (i_{\max} j_{\max}) \rceil, \end{cases} \quad (4)$$

где r – разрядность вычислений, μ – параметр цикла (число каналов радиовысотомера), i, j – параметры цикла (размерность матрицы т.н. гипотезной сетки [13]). Верхние индексы (1) и (2) обозначают циклическую и разовую части алгоритма соответственно. Анализ (4) показал, что архитектура ФОП практически полностью будет определяться циклической частью алгоритма привязки.

На рис. 3 приведена архитектура одного из разработанных SIMD-ФОП для реализации алгоритма привязки, она может служить основой разработки параллельных сопроцессоров для суперкомпьютеров.

Анализ аппаратных затрат в вентилях типа 2И-НЕ показал, что данный ФОП может быть реализован в виде кремниевого кластера СБИС (примерно 10% площади кристалла размером 10×10 мм² для технологии CMOS с нормами 90 нм).

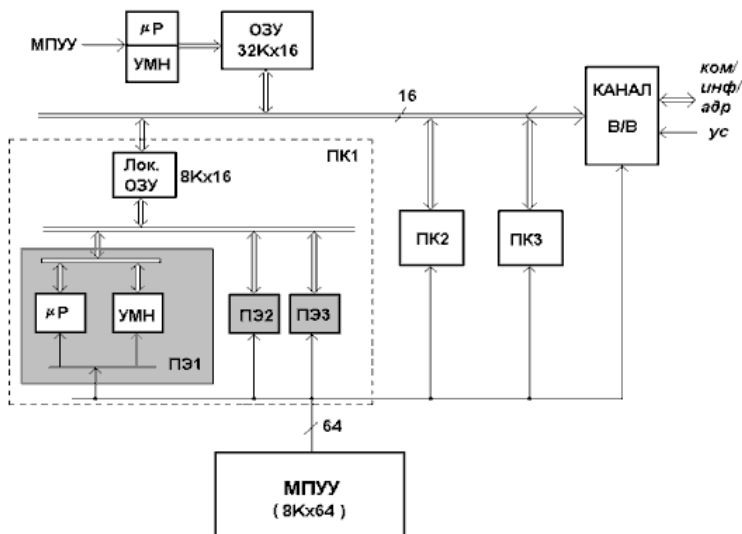


Рис. 3. Архитектура SIMD-ФОП

Высокопроизводительные ФОП на основе однородных вычислительных сред (ОВС-ФОП)

Реализация массового параллелизма обработки данных, обеспечиваемая ОВС, теоретически позволяет обеспечить предельную производительность при решении конкретных вычислительных задач. В течение последних 20 лет ОВС вновь служат объектом исследований и разработок в ряде стран и организаций, в ряде случаев имеются положительные результаты практической реализации ОВС [16], [17].

В течение последних 15 лет усилиями сотрудников Группы микропроцессорных архитектур ИМАШ УрО РАН в кооперации с ведущими приборостроительными предприятиями России (в первую очередь, НПО автоматики им. ак. Н.А. Семихатова, Екатеринбург) был разработан ряд ОВС-ФОП для реализации сложных вычислительных алгоритмов в реальном времени:

а) ОВС-ФОП на базе геометрико-арифметического параллельного процессора (ГАПП, аналог – процессор GAPP разработки фирмы NCR, USA). Основой этого ФОП является систолическое операционное устройство (СОУ), которое представляет собой 2D-массив процессорных элементов (ПЭ), связанных между собой локальными связями. Каждый ПЭ содержит битовое АЛУ, систему коммутаторов и ОЗУ емкостью 128 бит. Была разработана архитектура заказной СБИС СОУ размерностью 8×8 ПЭ. Разработка технологии СБИС

СОУ была проведена во ФГУП НПЦ "Элвис", а изготовление – на ПО "Ангстрем" (Зеленоград, Россия). Для обеспечения минимальных массы и габаритов ОВС-ФОП была разработана однослойная ситалловая печатная микроплата размером $60 \times 48 \text{ мм}^2$, на которой размещается 8 бескорпусных кристаллов СБИС, что позволяет иметь 512 ПЭ на одну плату. Разводка шин питания, адресов локальных ОЗУ и синхронизирующих импульсов осуществлялась на каждой плате отдельно, это обеспечивало соединение всех плат между собой только через внешние контакты.

Для технологических норм 45 нм CMOS-технологии возможно реализовать 2D-массив ПЭ размерностью 128×128 в одном кристалле. Это открывает перспективу использования ОВС-ФОП в качестве сопроцессоров в составе процессорных ядер общего назначения, персональных компьютеров и суперЭВМ;

б) ОВС-ФОП с архитектурой MIMD. Начиная со второй половины 1990-х годов кооперацией предприятий "Суперкомпьютерные системы (СКС)" (Москва), ИМаш УрО РАН и НПО автоматики (Екатеринбург) проводились совместные работы в российской части проекта СКИФ. Предприятием СКС была разработана СБИС «Minitera» [18], архитектура которой развивает концепцию MIMD-ОВС. Каждая СБИС содержала 25 ПЭ и работала на частоте 30 МГц. В ИМаш УрО РАН были проведены исследования реализации на ОВС Minitera быстрых алгоритмов умножения и деления над числами произвольной разрядности, параллельной сортировки в больших массивах чисел, цифровой обработки точечных изображений, некоторых алгоритмов криптографии.

Особенностью MIMD ОВС-ФОП является то, что всегда найдется такое значение параметров алгоритмов, начиная с которого время вычисления на MIMD ОВС-ФОП будет всегда меньше, чем для известных процессорных платформ. Была показана высокая эффективность ОВС данного типа при реализации широкого круга задач систем реального времени. В связи с этим можно рассматривать MIMD-ОВС в качестве аппаратной основы для создания высокопроизводительных сопроцессоров в ВС различного назначения – от бортовых ЦВМ до суперкомпьютеров.

1. Лукин Н.А. Основы теории проектирования архитектур функционально-ориентированных процессоров для систем реального времени // Высокопроизводительные вычислительные системы. Материалы Пятой Международной научной молодежной школы и Пятой Международной молодежной научно-технической конференции, 31 августа – 6 сентября 2008, Таганрог. – Таганрог: Изд-во ТТИ ЮФУ, 2008. – С. 115 – 166.

2. J. Cui, J. L. Bordim, K. Nakano, T. Hayashi, N. Ishii. Multithreaded Parallel Computer Model with Performance Evaluation//Workshop on Advances in Parallel and Distributed Computational Models. IPDPS 2000 Workshop.
3. Cristina Nicolescu and Pieter Jonker. Parallel low-level image processing on a distributed-memory system// Parallel and Distributed Computing in Image Processing, Video Processing, and Multimedia. IPDPS 2000 Workshop
4. http://en.wikipedia.org/wiki/Embedded_Supercomputing
5. Reddi V. Supercomputer Performance on a Chip Powers Next Generation Embedded Image Processing. <http://rtcmagazine.com/articles/view/102184>
6. Функционально ориентированные процессоры/ А.И. Водяхо, В.Б. Смолов, В.У. Плюснин, Д.В. Пузанков; Под ред. В.Б.Смолова. - Л.: Машиностроение. Ленингр. отд-ние, 1988.
7. Лупанов О.Б. Курс лекций по дискретной математике. Изд-во МГУ, 2006 г., 38 с.
8. Колмогоров А.Н. Теория информации и теория алгоритмов. Москва, Наука, 1987 г. С. 199 – 203.
9. Карацуба Е.А. Быстрые алгоритмы и метод БВЕ, <http://www.ccas.ru/personal/karatsuba/alg.htm>.
10. Попов Б.А. Вычисление функций на ЭВМ. - К.: Наукова Думка, 1984.
11. Лукин Н.А. Архитектурный синтез функционально-ориентированных процессоров математических функций// Гироскопия и навигация, 2003, №3. С. 109 – 120.
12. Лукин Н.А. Параллельные алгоритмы вычисления математических функций и их количественные характеристики // Новые информационные технологии в исследовании дискретных структур. Доклады второй всероссийской конференции, Екатеринбург, УрО РАН, 1998. - С.157-172.
13. Лукин Н.А. Системное проектирование функционально-ориентированных процессоров для бортовых корреляционно-экстремальных навигационных систем // Вестник Самарского государственного аэрокосмического университета имени академика С. П. Королёва. - 2009. - № 4 (20). - С. 218-236.
14. Воеводин В.В. Математические модели и методы в параллельных процессах. - М.: Наука, 1986.
15. Лукин Н.А. Функционально-ориентированные процессоры с однородной архитектурой для реализации алгоритмов бортовых систем управления // Высокопроизводительные вычислительные системы. Параллельные вычисления и задачи управления РАСО'2010: Труды V международной конференции, г. Москва, 26-28 октября 2010 г. - Институт проблем управления им. В.А. Трапезникова РАН. - С. 1177-1185.
16. Богачев М.П. Архитектура вычислительной системы с однородной структурой. В кн. Однородные вычислительные среды. Львов. ФМИ АН УССР. 1981.
17. Шмойлов В.И. Пульсирующие информационные решетки и суперкомпьютеры класса А. - Львов: Меркатор, 2005.
18. http://www.minitera.ru/rus/win/pbm_proc_arh.html

Д.А. Михайлин¹, Д.А. Тучин², В.В. Фитенко¹, В.С. Ярошевский²

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ДОПЛЕРОВСКОГО ИЗМЕРИТЕЛЯ СКОРОСТИ И ДАЛЬНОСТИ

¹ОАО «Концерн радиостроения «Вега», г. Москва,

²Институт прикладной математики им. М.В. Келдыша, г. Москва,
tau_301@mail.ru, phobos.den@yandex.ru

Решена математическая задача определения дальности до поверхности с помощью доплеровских измерений на нескольких частотах. Сформулированы условия определения однозначной дальности.

Доплеровский измеритель скорости и дальности (ДИСД) обеспечивает обнаружение и радиолокационный захват сигнала, отраженного от подстилающей поверхности. В отечественных проектах мягкой посадки на поверхность Луны ДИСД обеспечивает:

- измерение четырех проекций вектора скорости КА относительно поверхности Луны на лучи антенн;
- измерение четырех наклонных дальностей до поверхности Луны по лучам антенн;
- определение двух углов отклонения нормали к поверхности Луны от направления оси ОХ приборной системы координат.

Прибор способен измерить неоднозначные наклонные дальности B_i с точностью до длины излучаемой полуволны сигнала L_i . Задача состоит в восстановлении дальности D до поверхности при известных неоднозначных дальностях на двух разных частотах излучаемых волн. Для двух излучаемых частот справедливо:

$$\begin{cases} D_1 = n_1 \cdot L_1 + B_1 \\ D_2 = n_2 \cdot L_2 + B_2 \end{cases},$$

где D_1 и D_2 – однозначные дальности, которые, в общем случае, при наличии шумов и флуктуационной составляющей не будут равны; n_1 и n_2 – целые числа полуволн. Проблема сведена к решению системы уравнений в целых числах.

Разработан алгоритм восстановления однозначной дальности по измерениям на разных частотах модуляции. Статистическое моделирование работы алгоритма показало безошибочность его работы в условиях помех.

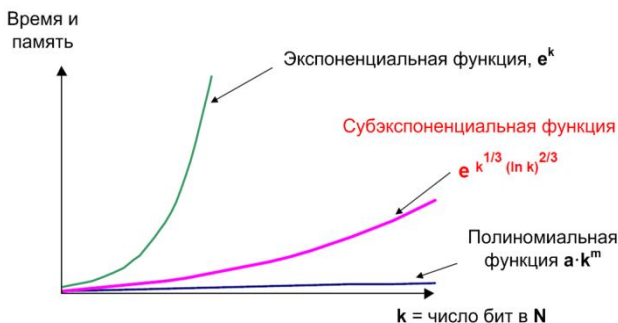
Математическое моделирование работы ДИСД показывает, что при принятых моделях измерителя прибор способен решать задачи при посадке на Луну.

РЕШЕНИЕ ЗАДАЧИ ФАКТОРИЗАЦИИ БОЛЬШИХ ЧИСЕЛ МЕТОДОМ ЭЛЛИПТИЧЕСКИХ КРИВЫХ НА РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
pelipeis@mail.ru, k.a.g@bk.ru*

Разложение целых чисел на простые множители (факторизация) является актуальной задачей в современных научных областях, использующих точные символьные вычисления с большими числами. Согласно теории алгоритмов, факторизация относится к классу вычислительно сложных задач, требующих для своего решения значительные вычислительные ресурсы: процессорное время и память [1].

Все существующие алгоритмы факторизации, в зависимости от вычислительной сложности, принадлежат к двум основным группам. К первой группе относятся экспоненциальные алгоритмы (решето Эратосфена, метод Ферма, ρ -алгоритм Полларда и др.), ко второй – субэкспоненциальные алгоритмы (метод квадратичного решета, метод эллиптических кривых и др.). У перечисленных групп зависимость сложности от длины факторизируемого числа N представлена, соответственно, экспоненциальной и субэкспоненциальной функциями (см. рисунок). Алгоритмы факторизации с полиномиальной сложностью в настоящее время неизвестны.



Вычислительная сложность алгоритмов факторизации

Среди субэкспоненциальных алгоритмов наиболее подходящим для реализации на реконфигурируемых вычислительных системах (РВС) [2] является метод эллиптических кривых (Elliptic Curve Method, ECM) [3]. Это обусловлено меньшим количеством необходимой памяти

и операций ввода-вывода по сравнению с другими алгоритмами, а также тем, что базовые вычислительные операции ЕСМ легко поддаются распараллеливанию.

Существующие реализации метода эллиптических кривых на реконфигурируемых вычислительных системах известны с 2005 года [4-6], однако всем им присущ ряд следующих существенных недостатков, ограничивающих реальную производительность многопроцессорных систем при решении задач факторизации:

- фиксированная разрядность k -бит факторизуемого числа N . Для факторизации чисел разрядности, отличной от k , необходима реконфигурация вычислительной системы;

- интенсивный обмен данными РВС с хост-компьютером на этапах вычислений между двумя фазами алгоритма ЕСМ [7]. Расчет результата первой фазы, принятие решения о продолжении вычислений, а также подготовка и загрузка исходных данных второй фазы алгоритма происходят после остановки вычислительного процесса первой фазы, выгрузки результатов и их обработки хост-компьютером;

- структурная реализация второй фазы алгоритма. Первая и вторая фазы алгоритма ЕСМ представлены в виде отдельных структур, что приводит к значительному увеличению использования логических ресурсов ПЛИС.

С целью устранения перечисленных недостатков предыдущих реализаций ЕСМ был выработан и применен ряд аппаратно-программных решений для оптимизации алгоритма, начиная от базовых операций модулярной арифметики и заканчивая организацией вычислительного процесса на уровне взаимодействия РВС и хост-компьютера. Полученная в результате реализация обладает следующими характеристиками:

- разрядность факторизуемого числа может находиться в диапазоне $k=512\div 992$ бита с шагом 32. Реализовано гибкое масштабирование базовых вычислителей – модулярного умножителя Монтгомери и модулярного сумматора-вычитателя;

- отсутствие паузы в работе РВС между двумя фазами алгоритма. Инициализация работы включает в себя загрузку исходных данных не только для первой, но и для второй фазы, которая выполняется всегда, без принятия решения со стороны хост-компьютера;

- микропрограммная реализация второй фазы алгоритма за счет использования аппаратных ресурсов ПЛИС для первой фазы. Функциональные ресурсы первой фазы вместе с автоматом управления и циклическим буфером памяти для хранения микропрограмм позволяют реализовать универсальный вычислитель ЕСМ, архитектурно не зависящий от особенностей информационной структуры второй фазы.

В таблице представлены результаты реализации ЕСМ на вычислительном модуле (ВМ) «Тайгета». ВМ «Тайгета» представляет собой реконфигурируемую систему, состоящей из 32 ПЛИС фирмы Xilinx семейства Virtex 7, каждая из которых содержит 48,5 млн. эквивалентных вентилей. В каждой ПЛИС было реализовано 28 вычислительных блоков ЕСМ. Тактовая частота работы вычислительной схемы составила 200 МГц. Сравнение проводилось с персональным компьютером на базе процессора Intel Core i-5 3,4 ГГц, 16 ГБ ОЗУ.

Результаты реализации ЕСМ на РВС «Тайгета»

$b1, b2 (D=1024) \setminus N$		512	767	992
$b1=10000$ $b2=1000000$	Ускорение, вычислительное	444,836	596,648	885,253
	Ускорение, общее	291,714	350,315	475,192
$b1=50000$ $b2=50000000$	Ускорение, вычислительное	409,435	547,891	865,523
	Ускорение, общее	388,110	518,512	822,227
$b1=100000$ $b2=100000000$	Ускорение, вычислительное	391,360	545,798	853,941
	Ускорение, общее	350,934	492,985	784,712

Параметры-ограничители $b1$ и $b2$ задают границы гладкости факторной базы для первого и второго этапов, D – параметр, определяющий соотношение между временем вычислений и объемом необходимой памяти, N – разрядность факторизуемого числа. В таблице приведены данные для двух видов ускорения – вычислительного и общего. Вычислительное ускорение учитывает только время работы вычислительной схемы. В расчете же общего ускорения учитывались также время подготовки данных, расчет необходимых параметров и постобработка.

На основании анализа данных таблицы можно отметить, что ускорение как вычислительное, так и общее, растет с увеличением разрядности факторизуемого числа. Кроме того, тесты показали, что с увеличением числа кривых время предварительных вычислений оказывает меньшее влияние на общее время работы, и, как следствие, растет об-

шая производительность системы, что позволяет сделать вывод об эффективности предлагаемого подхода к реализации ЕСМ на РВС.

1. Ишмухаметов Ш.Т. Методы факторизации натуральных чисел: Учебное пособие. – Казань: Казан. ун., 2011. – 190 с.
2. Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И. Реконфигурируемые мультиконвейерные вычислительные структуры / Изд. 2-е, перераб. и доп. / Под общ. ред. И. А. Каляева. Ростов н/Д: Издательство ЮНЦ РАН, 2009. – 344 с.
3. Lenstra H.W. Factoring integers with elliptic curves / H.W. Lenstra.- Ann.Math. v.126 (1987), p. 649-674.
4. M. Šimka, J. Pelzl, T. Kleinjung, J. Franke, C. Priplata, C. Stahlke, M. Drutarovský, V. Fischer, and C. Paar, “Hardware Factorization Based on Elliptic Curve Method” // Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2005), J. Arnold and K. L. Pocek, Eds. IEEE Computer Society, April 18-20 2005, pp. 107-116..
5. K. Gaj, S. Kwon, P. Baier, P. Kohlbrenner, H. Le, M. Khaleeluddin, and R. Bachimanchi, “Implementing the Elliptic Curve Method of Factoring in Reconfigurable Hardware” // Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems (CHES 2006), ser. LNCS, L. Goubin and M. Matsui, Eds., vol. 4249. Springer-Verlag, 2006, pp. 119-133.
6. R. Zimmermann, T. Güneysu, C. Paar, “High-Performance Integer Factoring with Reconfigurable Devices ” // Field-Programmable Logic and Applications (FPL 2010), IEEE, pp. 83-88 .
7. Montgomery P.L. Speeding the Pollard and Elliptic Curve Methods of Factorization./ P.L. Montgomery // Mathematics of Computation, v.48, iss.177,1987, p.234-264.

С.А. Степаненко

ОЦЕНКИ УСКОРЕНИЯ ВЫЧИСЛЕНИЙ ГИБРИДНЫМИ РЕКОНФИГУРИРУЕМЫМИ СИСТЕМАМИ

*Российский федеральный ядерный центр
«ВНИИ экспериментальной физики», г. Саров,
ssa@vniief.ru*

Введение

В [1] получены оценки ускорения вычислений гибридными системами, содержащими универсальные процессоры (MIMD-компонент) и арифметические ускорители (SIMD-компонент), по сравнению с однородными системами из универсальных процессоров.

Здесь исследуются значения ускорения, которые могут быть достигнуты для заданного вычислительного процесса в результате реконфигурации структуры гибридной системы, содержащей фиксированное количество процессоров и ускорителей.

В отличие от реконфигурируемых вычислителей общего вида, используемых в [2,3], рассматриваемые ниже системы содержат лишь MIMD и SIMD компоненты. Ограничение типов компонентов позволяет существенно упростить декомпозицию и реконфигурацию.

Реконфигурация состоит в изменении количества процессоров и количества взаимодействующих с ними ускорителей в рамках имеющихся в системе.

Эта работа развивает и дополняет [1, 4]. В ней получены оценки ускорения, которые можно достигнуть реконфигурируемой гибридной вычислительной системой.

Создаваемые гибридные вычислительные системы [5] содержат десятки тысяч процессоров и ускорителей; в перспективе эти количества возрастут. Актуальность этой работы обусловлена необходимостью достижения наибольшей производительности и полного задействования ресурсов.

Основные определения

Гибридный вычислитель содержит q универсальных процессоров, реализующих MIMD вычисления, и $г$ арифметических ускорителей; каждый ускоритель реализует SIMD вычисления. Структура гибридного вычислителя показана на рис. 1.

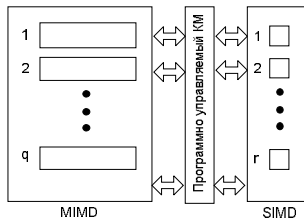


Рис. 1. Структура реконфигурируемого вычислителя

В качестве MIMD компонента могут применяться вычислительные модули, содержащие универсальные одноядерные или многоядерные процессоры, SMP системы или MPP системы [6].

Примерами SIMD компонентов являются арифметические ускорители фирм NVIDIA и AMD, процессоры Cell, ClearSpeed и т.п.[5, 7].

Примером коммутатора является PEX 8696, позволяющий коммутировать 1, 2, 4, 8 и 16 ускорителей с несколькими (до восьми) вычислительными модулями [7].

Пусть решение задачи одним ядром универсального процессора требует интервал длительностью T_1 .

Полагаем, что процесс решения этой же задачи гибридным вычислителем, содержащим один процессор и один ускоритель, занимает интервал длительностью

$$T_{i,l} = T_M + T_S,$$

где $T_M = T_1 \varphi$ – длительность вычислений, выполняемых процессором;

$0 \leq \varphi \leq 1$ – доля вычислительного процесса, выполняемого универсальным процессором (доля MIMD фрагмента);

$T_S = (1 - \varphi) \frac{T_1}{\rho}$ – длительность вычислений, выполняемых ускорителем;

$\rho > 1$ – коэффициент ускорения по сравнению с универсальным процессором, достигаемый применением ускорителя (на SIMD фрагменте).

Гибридный вычислитель из одного ядра и одного ускорителя называем *элементарным вычислителем*, а выполняемый им процесс – *элементарным процессом*.

Значения параметров ρ и φ измеряются на элементарном вычислителе, содержащем одно ядро и один ускоритель. Они называются первичными параметрами.

Оценки ускорения вычислений реконфигурируемыми гибридными средами

Пусть имеется вычислительная среда $S_{q,q}$, содержащая q ядер и q ускорителей.

В [1] установлено, что вычислительный процесс с первичными параметрами φ и ρ может быть выполнен этой системой для распараллеливания в режиме умножения задачи (закон Густафсона [6]) в течение интервала

$T_{q,1} = T_1 \varphi + T_1 (1 - \varphi) \frac{q}{\rho}$, если задействованы q ядер и 1 ускоритель, либо

$T_{1,q} = T_1 \varphi \cdot q + T_1 (1 - \varphi) \frac{1}{\rho}$, если задействованы 1 ядро и q ускорителей, либо

$T_{q,q} = T_{1,1} = T_1 \varphi + T_1 (1 - \varphi) \frac{1}{\rho}$, если задействованы q ядер и q ускорителей.

Соответствующие коэффициенты ускорения вычислений по сравнению с одним ядром имеют вид

$$K_{q,1} = \frac{q}{\varphi + (1 - \varphi) \frac{q}{\rho}}; K_{1,q} = \frac{q}{\varphi \cdot q + (1 - \varphi) \frac{1}{\rho}}; K_{q,q} = \frac{q}{\varphi + \frac{1 - \varphi}{\rho}}.$$

Наиболее «быстро» растущим является $K_{q,q}$, когда одновременно задействованы q ядер и q ускорителей; этот вариант назовем *линейным*. Однако в первых двух вариантах остаются свободными и могут быть

задействованы для решения других задач 1 ядро и $(q - 1)$ ускорителей в первом варианте и $(q - 1)$ ядер и 1 ускоритель во втором.

Оценим возможности одновременного полного задействования оборудования для этих вариантов и достижения большего по сравнению с $K_{q,q}$ ускорения вычислений, тем самым оценим целесообразность применения средств реконфигурации.

Потребуем, чтобы вычислительная среда одновременно выполняла два вычислительных процесса; один задействует $(q - 1)$ ядер и 1 ускоритель, другой 1 ядро и $(q - 1)$ ускорителей. Полагаем, что $T_{(q-1),1} = T_{1,(q-1)} = T_R$.

Повторим их q раз. В результате в течение интервала qT_R будет выполнено $2q(q - 1)$ элементарных вычислительных процессов.

Эти же вычислительные процессы потребуют интервала $2(q - 1)T_{q,q}$, если их выполнять, задействуя одновременно q ядер и q ускорителей, имеющих в гибридной системе.

Применение реконфигурации позволяет ускорить вычисления, если достигается: $qT_R < 2(q - 1)T_{q,q}$.

Последнее выполняется, если

$$q < 1 + \frac{\varphi\rho}{1 - \varphi}. \quad (1)$$

Значение ускорения, достигаемое применением реконфигурации, можно оценить отношением

$$K_R = \frac{2(q - 1)T_{q,q}}{qT_R} = 2(1 - \frac{1}{q})C, \text{ где } C = \frac{\varphi + \frac{1 - \varphi}{\rho}}{\varphi + \frac{1 - \varphi}{\rho}(q - 1)} < 1. \quad (2)$$

Таким образом, для заданного q , удовлетворяющего условию (1), ускорение, достигаемое рассматриваемой реконфигурацией, не превосходит 2 по отношению к $K_{q,q}$, т.е. когда в системе полностью задействованы все ядра и все ускорители.

По сравнению с одним ядром процессора ускорение, достигаемое реконфигурируемой системой, составляет $R_{q,q} = K_{q,q} \cdot K_R$, т.е. $R_{q,q} = 2 \frac{q - 1}{\varphi + \frac{1 - \varphi}{\rho}(q - 1)}$.

Тем самым подтверждается возможность существенного, но не более чем вдвое, роста ускорения по сравнению с нереконфигурируемыми статическими структурами.

Полученные оценки ускорения справедливы и для распараллеливания в режиме «деления» размера задачи (закон Амдала [6]).

Иллюстрации применения соотношений

Проиллюстрируем полученные соотношения.

Пусть вычислительный процесс характеризуется следующими значениями: $T_l = 100$ сек, $T_{l,l} = 21,6$ сек, $T_M = 20$ сек и $T_S = 1,6$ сек. Тогда $\varphi_l = 0,2$, $\rho_l = 50$.

Значения $K_{q,l}$ и $K_{l,q}$ для этого процесса приведены на рисунке 2. Эффект реконфигурации достигается, если $q < 1 + \frac{\varphi_l \rho_l}{1 - \varphi_l} \leq 13$.

Для $q = 3$, $q = 5$ и $q = 10$ получаем $R_{3,3} = 17,1$, $R_{5,5} = 30$ и $R_{10,1} = 52,2$.

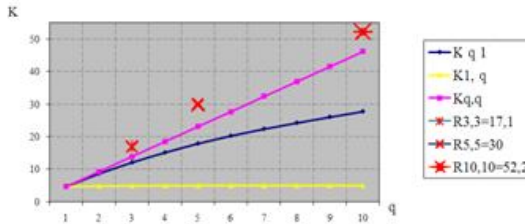


Рис. 2. Значения коэффициентов ускорения вычислений

Заключение

Полученные соотношения показывают, что при выполнении определенных условий применение реконфигурации позволяет по сравнению с нереконфигурируемыми структурами существенно, но не более чем вдвое, ускорить процесс вычисления гибридными системами.

Производителями вычислительных средств освоено выпуск систем, структурно идентичных исследованным в этой работе. Применение предложенных средств позволяет повысить эффективность вычислительных систем.

1. Степаненко С.А. Оценки ускорения вычислений гибридными системами // Пленарные доклады Пятой международной конференции <Параллельные вычисления и задачи управления> РАСО 2010 Москва 26-28 октября 2010г. М.: Учреждение Российской академии наук Институт проблем управления им. В.А.Трапезникова РАН. С.61-71.
2. Каляев И.А., Левин И.И. и др. Реконфигурируемые мультимониторные вычислительные структуры. Ростов-на-Дону: ЮНЦ РАН, 2009.-344 с.
3. Bondalapati K., Prasanna V.K. Reconfigurable Computing Systems. Proc. IEEE 2002, 90, № 7. P. 1201-1217.
4. Степаненко С.А. «Способ определения структуры гибридной вычислительной системы». Патент Российской Федерации № 2436151 на изобретение.
5. Oak Ridge changes Jaguar's spots from CPUs to GPUs [Электронный ресурс]. Режим доступа: http://www.theregister.co.uk/2011/10/11/oak_ridge_cray_nvidia_titan/
6. Цилькер Б.Я., С.А.Орлов. Организация ЭВМ и систем. С.-Пб., 2004г.
7. [Электронный ресурс]. Режим доступа: <http://content.dell.com/us/en/gen/spredir.ashx/solutions/poweredge-c410x-thinkers>

**ПРИМЕНЕНИЕ ЯЗЫКА OPENCL
ПРИ РАЗРАБОТКЕ СИСТЕМ НА КРИСТАЛЛЕ В FPGA
ФИРМЫ ALTERA**

*НИЦ (г. Курск) ФГУП «18 ЦНИИ» МО РФ,
г. Курск,
nevgeney@gmail.com*

Основными подходами, применяемыми для удовлетворения потребности во всевозрастающей производительности вычислительных комплексов различного назначения, являются следующие. Первый заключается в увеличении рабочей частоты вычислительных элементов комплекса. Недостатком данного подхода является невозможность увеличения рабочей частоты с сохранением постоянной плотности напряжения. Суть второго подхода сводится к созданию сложных аппаратных средств, которые могли бы обеспечивать параллелизм на уровне команд. В этом случае значительная часть аппаратных ресурсов процессора расходуется на преобразование последовательного кода в параллельно выполняемый, а также для компенсации задержек памяти, т.е. неэффективно.

Преодоление указанных проблем возможно на путях создания систем автоматического извлечения параллелизма на уровне команд во время их выполнения, обеспечивающих высокую производительность и низкое энергопотребление. Наиболее перспективным представляется построение указанных систем на базе FPGA. В то же время, процесс разработки приложений на базе FPGA является весьма трудоемким вследствие необходимости перебора множества вариантов отображения пользовательских алгоритмов на внутреннюю структуру ПЛИС и сложных процедур отладки.

Одним из способов снижения сложности и сокращения времени разработки систем на кристалле для FPGA фирмы Altera является применение языка OpenCL, который позволяет программисту явно указывать и контролировать параллелизм, создавая прикладные программы, более соответствующие параллельной природе ПЛИС, чем последовательные программы, описанные на других языках высокого уровня.

На рис. 1 представлена традиционная для ПЛИС фирмы Altera OpenCL-система, содержащая несколько конвейерных ядер и схему подключения конвейеров к интерфейсу внешней памяти. В дополнение к ядру конвейера компилятор Altera OpenCL также создает интерфейсы к внешней и внутренней памяти.

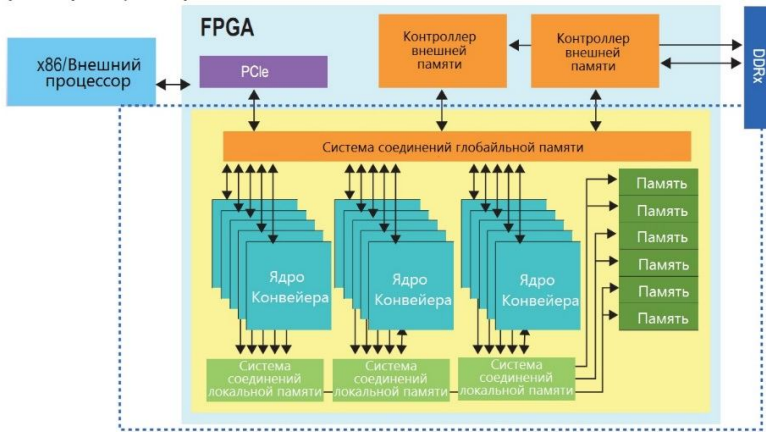


Рис. 1. Система OpenCL на ПЛИС фирмы Altera

Модули, отвечающие за хранение и загрузку данных для каждого конвейера, подключены к внешней памяти через глобальную систему соединений, которая отвечает за распределение запросов к группе памяти DDR DIMM. Доступ к локальной памяти OpenCL осуществляется через специализированную систему соединительных линий FPGA, обеспечивающую высокую частоту и эффективную организацию запросов, к блокам памяти M9K.

Традиционно OpenCL-приложения состоят из двух частей: программы ядра и программы хоста (рис. 2).

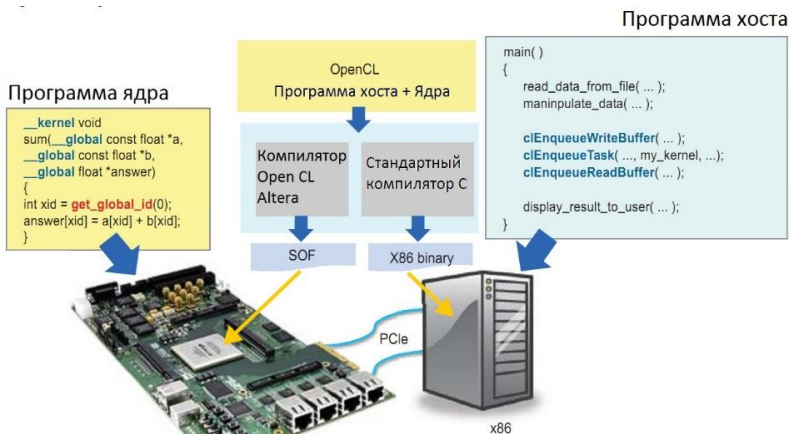


Рис. 2. Архитектура приложения OpenCL

Программа хоста описывается на языке C/C++ и может работать на любом виде микропроцессоров, в качестве которых могут использоваться встроенный процессор FPGA, аппаратный ARM процессор или внешний процессор x86.

Программа ядра может быть реализована на многоядерных процессорах, графических ускорителях, ПЛИС и т.д. Доступ программы хоста к реализациям ядер осуществляется через интерфейс прикладного программирования, обеспечивающий обмен и обработку данных на FPGA.

Основополагающим элементом OpenCL-to-FPGA компилятора является понятие конвейерного параллелизма, которое проиллюстрируем следующим примером.

Предположим, что компилятор создал три конвейера для программы ядра (рис. 3). На первом шаге работы системы поток 0 осуществляет тактирование двух модулей загрузки, что позволяет осуществить загрузку первых элементов данных из массивов A и B. На втором шаге поток 1 тактируется так же, как и поток 0, который завершил процесс чтения из памяти и сохранил результаты в регистрах в соответствии с использованными блоками загрузки. На третьем шаге осуществляется тактирование потока 2, поток 1 захватывает возвращаемые данные, а поток 0 сохраняет сумму двух значений, которые были загружены. Отметим, что в стационарном состоянии все части конвейера активны, при этом обработка каждого нового элемента осуществляется в отдельном потоке.

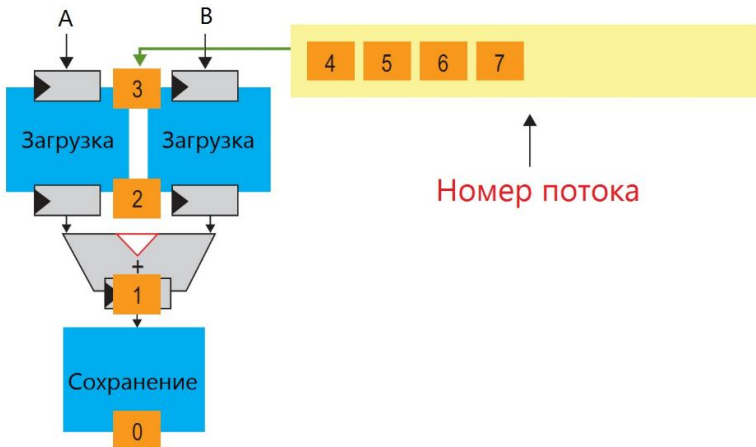


Рис. 3. Конвейеры ядра

Процесс создания проектов для FPGA с использованием OpenCL обладает рядом преимуществ по сравнению с традиционной методикой

проектирования на основе языков описания аппаратуры HDL. Как видно из рис. 4, разработка программируемых устройств включает следующие этапы: формулирование идеи, описание алгоритма на языке высокого уровня, например С, генерация при помощи компилятора потока команд.

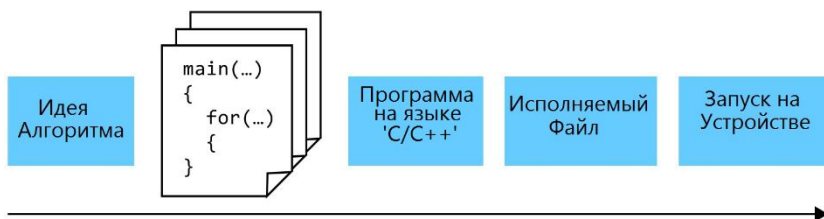


Рис. 4. Проектирование системы с использованием стандарта OpenCL

Основной задачей компилятора OpenCL является обеспечение возможности работы проектировщика только над описанием алгоритмов функционирования создаваемой системы, а не на тонкостях проектирования аппаратных систем на конкретной элементной базе. Это достигается за счет автоматизации выполнения как минимум следующего набора действий:

- описание информационных потоков;
- описание автоматов управления информационными потоками;
- создание системы соединений низкоуровневых IP-ядер;
- определение временных ограничений, накладываемых на внешние интерфейсы.

Применение описанной методологии проектирования цифровых устройств позволяет разработчику с легкостью переходить на новые FPGA, которые имеют большую производительность и емкость кристалла. Данный факт обусловлен возможностями компилятора OpenCL, который, с использованием дополнительных библиотек, преобразует один и тот же высокоуровневый код на языке C/C++ в аппаратные конвейеры, обеспечивающие применение дополнительных возможностей новых FPGA.

1. Altera corp. «Implementing FPGA Design with the OpenCL Standard».
2. Kavya S Shagritthaya «Enabling Development of OpenCL Applications on FPGA platforms», Blacksburg, Virginia, 2012.

РЕКОНФИГУРИРУЕМЫЙ ВЫЧИСЛИТЕЛЬНЫЙ МОДУЛЬ НА БАЗЕ ПЛИС CYCLONE V

*НИЦ (г. Курск) ФГУП «18 ЦНИИ» МО РФ, г. Курск,
nevgeney@gmail.com*

Для решения высокопроизводительных и трудоемких задач широко используются параллельные вычислительные системы на базе ПЛИС. Главными достоинствами этих систем являются не только высокая производительность, но и модульность, компактность составных частей, простота масштабируемости и высокие коммуникационные возможности. Целью данного исследования является создание реконфигурируемого модуля, удовлетворяющего требованиям модульности, компактности и простоты масштабируемости.

В докладе приведено описание реконфигурируемого вычислительного модуля на базе ПЛИС Cyclone V, предназначенного для решения трудоемких вычислительных задач цифровой обработки сигналов и битовой обработки данных.

Структурно-функциональная схема вычислительного модуля на базе ПЛИС Cyclone V представлена на рис. 1.

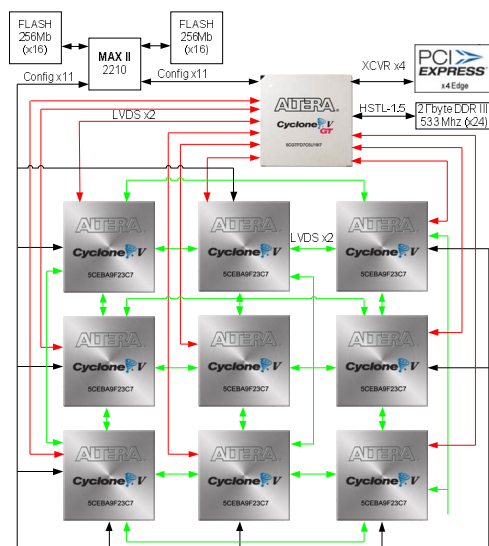


Рис. 1. Структурно-функциональная схема реконфигурируемого вычислительного модуля на базе ПЛИС Cyclone V

Вычислительный модуль состоит из следующих компонентов:

- девять ПЛИС 5CEBA9F23C7 семейства CycloneV, используемых в качестве платформы для реализации пользовательских алгоритмов обработки данных (вычислительные микросхемы);
- одна ПЛИС 5CGTFD7C5U19I7 семейства CycloneV, используемая в качестве контроллера вычислительной части;
- одна микросхема Max II 2210, используемая для конфигурации вышеописанных ПЛИС;
- две микросхемы Flash-памяти объемом 256 Мб каждая, используемые для хранения конфигурационной последовательности ПЛИС;
- распределенная память типа DDRIII общим объемом 2 Гб с максимальной частотой записи/чтения 533 МГц, подключенная к ПЛИС, реализующей функции контроллера вычислительной части.

Каждая вычислительная микросхема соединена с каждой из четырех соседних, а также с микросхемой контроллера двумя каналами интерфейса LVDS с частотой до 800 МГц. Для организации межмодульного взаимодействия используется интерфейс PCIe Gen2 x4, обеспечивающий ввод/вывод данных со скоростями до 16 Гбит/с.

Некоторые технические характеристики используемых в конструкции модуля ПЛИС представлены в таблице 1.

Технические характеристики ПЛИС CycloneV

Таблица 1

Наименование	5CGTFD7C5U19I7	5CEBA9F23C7
Размеры, мм	19x19	23x23
Количество ALUT	112 960	227 120
Количество пользовательских контактов ввода/вывода	268	224
Количество трансиверов	6	0
Количество аппаратных ядер PCIExpress	1	0
Количество аппаратных ядер DDR	2	0
Количество модулей блоковой памяти (по 10240 бит)	686	1 220
Количество блоков цифровой обработки сигналов, сконфигурированных в 1 умножитель 27*27 или в 2 умножителя 18*19	156	342
Блок автоматической фазовой подстройки частоты (PLL)	7	8
Блок автоматической подстройки по задержке (DLL)	4	4
Ориентировочная стоимость, руб.	9 000	7 830

Суммарная логическая емкость вычислительного модуля составляет 2 157 040 LUT. Модуль содержит 119 459 840 бит блоковой памяти, работающей на частоте до 300MHz, а также 3234 блока цифровой обработки сигналов (один умножитель 27×27 бит или два умножителя 18×19 бит).

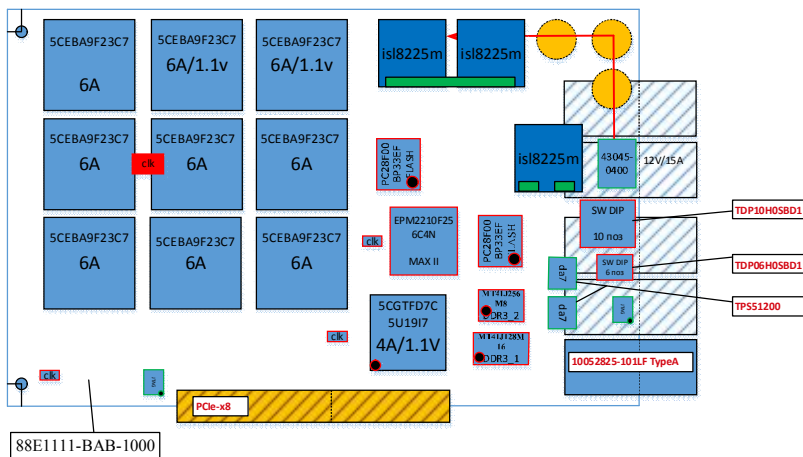


Рис. 2. Схема компоновки вычислительного модуля в форм-факторе PCIExpress

Результаты проведенного моделирования показывают, что ожидаемое расчетное значение производительности составит порядка $1,6 \cdot 10^{13}$ 32-битных операций в секунду. Ниже приведем сравнение модуля с существующими вычислительными модулями на примере задач обработки битовых последовательностей (таблица 2).

Таблица 2

Тип модуля	Количество обработанных слов/сек
Вычислительный модуль на базе ПЛИС Cyclone V	$162 * 10^9$
Модуль на основе GPU Nvidia GeForce GTX 295	$213,4 * 10^6$
Модуль на основе AMD Phenome 9950 Quad-Core 2.61 GHz	$10,4 * 10^6$
Модуль M200	$122,8 * 10^9$
Модуль PBC-0.2 ВБ	$512,7 * 10^9$
Модуль «Орион»	$512,7 * 10^9$

В соответствии с данной таблицей стоит отметить, что производительность вычислительного модуля на базе ПЛИС Cyclone V значительно превышает вычислительные возможности CPU и GPU, а также модуля M200, однако уступает модулям PBC-0.2 ВБ и «Орион». Но в отличие от модулей PBC-0.2 ВБ и «Орион» достаточно компактен.

1. Суперкомпьютерные технологии: разработка, программирование, (СКТ-2010) // Материалы Международной научно-технической конференции. Т. 1. – Таганрог: Изд-во ТТИ ЮФУ, 2010. – 340 с.

И.Е. Тарасов, Ю.И. Бялик, М.О. Кузелин

ОСОБЕННОСТИ СИСТЕМНОЙ АРХИТЕКТУРЫ СЕМЕЙСТВА FPGA XILINX ULTRASCALE

*ЗАО «КТЦ Инлайнгрупп»,
г. Москва,
ilya.tarasov@inlinegroup.ru*

Компания Xilinx в 2013 году объявила о начале выпуска очередного поколения FPGA, выполняемых с соблюдением 20-нм норм технологического процесса. Данное поколение представляет собой скорее эволюционное решение с точки зрения архитектуры основных программируемых модулей, однако предлагает совершенно иной подход к построению тактовых сетей, что требует внимания со стороны разработчиков. В состав нового поколения входят семейства Kintex UltraScale и Virtex UltraScale. Их характеристики представлены в таблице.

**Краткая сводка основных характеристик семейств Kintex UltraScale
и Virtex UltraScale**

Семейство	Kintex UltraScale	Virtex UltraScale
Логических ячеек, тыс.	355 – 1160	627 – 4407
Блочной памяти, Мб	19 – 75,9	44,3 – 115,2
Секций DSP	1700 – 5520	600 – 2880
Производительность DSP, GMAC/s	8180	4268
Приемопередатчиков	16 – 64	36 – 120
Пиковая скорость приемопередатчиков, Гбит/с	16	33
Пиковая производительность приемопередатчиков, Гбит/с	2086	5101
Интерфейсов PCIe	2 – 4	2 – 6
100G Ethernet	2	3 – 7
150G Interlaken	1	3 – 9
Производительность интерфейсов памяти, Мбит/с	2400	2400
Программируемых выводов	312 – 832	364 – 1456
Напряжение на выводах, В	1,0 – 3,3	1,0 – 3,3

Из приведенных сведений можно сделать некоторые выводы:

- диапазон логической емкости в очередной раз расширен, семейство Virtex как представитель верхнего сегмента получило микросхему емкостью 4,4 млн. логических вентилях (по сравнению с 2 млн. в Virtex-7);
- максимальная производительность в задачах цифровой обработки сигналов достигается при использовании семейства Kintex, в котором имеется до 5520 секций DSP (больше, чем в Virtex-7 или Virtex UltraScale);
- семейство Virtex позиционируется как платформа преимущественно для телекоммуникационных приложений (тем не менее, старшая микросхема не имеет большого количества приемопередатчиков и предназначена для прототипирования ASIC);
- произошли некоторые эволюционные изменения в количестве и скорости приемопередатчиков – до 104 шт. в Virtex, а скорости обмена составляют 16 и 33 Гбит/с;
- добавлены аппаратные блоки 100G Ethernet и 150G Interlaken, что позволяет в короткие сроки и с предсказуемыми характеристиками разрабатывать проводные телекоммуникационные системы на базе данных интерфейсов.

Положительной тенденцией для Xilinx можно считать регулярные модификации архитектуры FPGA, которые позволяют разработчикам получать преимущества не только от улучшения характеристик вследствие перехода к новому технологическому процессу, но и от появления новых компонентов на кристалле ПЛИС и улучшения характеристик уже имеющихся. При этом революционных изменений в архитектуре основных программируемых компонентов (логических ячеек, блочной памяти, блоков цифровой обработки сигналов) ожидать вряд ли приходится, поскольку их характеристики вполне соответствуют текущему уровню развития технологии и методов проектирования. Более важная проблема возникла в области проектирования на FPGA в течение последних лет с переходом от 90 нм (Virtex-4) к последующим поколениям технологических процессов и соответствующих им семейств FPGA. Проблема заключается в том, что по мере роста размеров кристалла количество логических ячеек возрастает пропорционально квадрату размера, а количество трассировочных линий растет всего лишь линейно. Поэтому по мере перехода к новым поколениям FPGA инженеры Xilinx регулярно увеличивали количество линий, проложенных между ячейками.

Очередной качественный скачок в этом направлении заявляется для семейств UltraScale. Количество разнообразных трассировочных ресурсов в очередной раз увеличивается, что в сочетании с применением новых алгоритмов «аналитического размещения» (analytical placement) призвано улучшить трассируемость проектов. Данное свойство представляется весьма важным в свете последних тенденций развития технологии производства цифровых интегральных схем. Можно заметить, что рост системной тактовой частоты замедлился, вплоть до того, что для семейств UltraScale заявлены те же значения рабочих частот основных программируемых ресурсов, что и для Virtex-7. Однако речь в данном случае идет не о частоте, на которой способен работать конкретный проект, представляющий практический интерес, а о несколько абстрактной схеме, состоящей из цепочки синхронных компонентов, с не более чем одной логической ячейкой между каждой парой регистров. Размещение связанных компонентов в такой цепочке должно быть компактным. Именно для такой конструкции определяется понятие «системная тактовая частота». На практике это значение можно смело полагать недостижимым и рассчитывать на меньшие значения, скорректированные с учетом сложности схемы и особенностей взаимного расположения компонентов.

Именно взаимное расположение компонентов является в настоящее время важнейшим фактором, определяющим достигаемую для проекта тактовую частоту. Еще для техпроцессов с нормами 90 нм было отмечено, что задержки на трассировочных линиях становятся сопоста-

вимыми с задержками на цифровых компонентах микросхем. Это является не следствием ухудшения характеристик трассировочных линий, а результатом повышения быстродействия цифровых компонентов. По мере уменьшения норм технологических процессов эффект не только сохранялся, но и усугублялся тем, что площадь кристалла (а значит, и длина трассировочных линий) постоянно увеличивалась. Поэтому повышение системной тактовой частоты при переходе на новые технологические процессы еще не означает, что этими возможностями удастся воспользоваться на практике. Не менее важно качественно улучшить характеристики трассировочных ресурсов. Именно на это и был сделан упор в новом поколении FPGA UltraScale.

Кроме передачи данных, для кристаллов большого размера стало необходимым также позаботиться о формировании тактовых сигналов. Уже в серии 7 намечаются некоторые проблемы с синхронизацией компонентов, расположенных в разных точках полупроводникового кристалла. Несмотря на то, что тактовое дерево (clocktree) реализовано с помощью выделенных трассировочных линий, при увеличении размеров FPGA инженеры Xilinx сталкиваются с вполне объективными препятствиями. На практике на тактовую частота проекта из сотен тысяч и даже миллионов логических ячеек будут влиять уже и характеристики джиттера и неравномерности распространения по кристаллу тактового сигнала. Эти показатели невозможно улучшить до бесконечности, тем более что увеличение размеров ПЛИС предъявляет все более и более жесткие требования к тактовым цепям. Одним из возможных решений для серии 7 является переход к т.н. GALS-архитектуре (Globally Asynchronous, Locally Synchronous), в рамках которой проект реализуется в виде нескольких подсистем, каждая из которых имеет собственный независимый источник тактового сигнала, располагаясь на кристалле компактно. Подсистемы не синхронизированы друг с другом (хотя номинальное значение их тактовых частот может и совпадать), а передача данных между ними происходит с помощью схем с аппаратной поддержкой двух синхронных интерфейсов, например *truedual-port RAM*.

Практика показывает, что корректное использование тактовых ресурсов FPGA до сих пор остается проблемой для некоторых разработчиков. Проблема заключается скорее в недостаточной актуализации проблемы формирования тактовых сигналов и необходимости использования для этой цели вполне определенных компонентов. Дополнительно ситуация усугубляется тем, что использование упрощенной схемы, состоящей только из глобального тактового буфера, само по себе не является гарантированным путем получения неработоспособного проекта, который мгновенно потребовал бы внимания разработчика и исправления ситуации. Источником проблемы в действительности является

ся объективная нестабильность временных характеристик современных цифровых микросхем, которую обычно иллюстрируют аббревиатурой PVT – Process, Voltage, Temperature. Понятие Process относится к вариациям технологического процесса, которые не позволяют изготовить абсолютно идентичные микросхемы. Поэтому проект, рассчитанный на конкретные значения задержек, будет работать только в определенных экземплярах FPGA. Термины Voltage и Temperature указывают на изменение задержек сигналов при колебаниях напряжения питания и температуры, что происходит постоянно в процессе эксплуатации устройств на базе ПЛИС. Влияние этих факторов становится все заметнее с уменьшением норм технологического процесса, поэтому уже начиная с 90-нм ПЛИС (Spartan-3, Virtex-4) разработчики часто сталкивались с явлением, когда логически корректный проект, проходящий требуемые функциональные тесты, оказывался неработоспособным либо при изменении условий эксплуатации, либо для отдельных микросхем при производстве партии изделий.

Важно еще раз подчеркнуть, что речь в данном случае не идет о недостатках ПЛИС как таковых или несовершенстве конкретного технологического процесса. Данные эффекты, по сути, являются прямым следствием повышения тактовой частоты, т.е. уменьшения ее периода. На фоне периода, измеряемого в единицах наносекунд, становятся более чем заметными влияние температурных эффектов, джиттер тактового сигнала и сами по себе времена распространения сигналов по кристаллу (размеры которого также растут с развитием технологии).

В устройствах семейств UltraScale применена новая схема формирования тактовых ресурсов, подобная тем, которые используются в заказных микросхемах (т.н. ASIC-like clocking). Схематичное изображение тактовых ресурсов ПЛИС UltraScale показано на рисунке.

Приведенная схема гораздо лучше соответствует подходу, основанному на разделении проекта на несколько тактовых регионов. Для каждой из подсетей, показанных на рисунке, можно установить собственный источник тактового сигнала, выбрав его среди ресурсов, размещенных в непосредственной близости (а не в центре кристалла, как в предыдущих поколениях FPGA). В каждом из тактовых регионов расположены 24 тактовых сети, которые могут тактироваться буферами, размещенными в том же регионе. Размер региона несколько увеличен по сравнению с серией 7 и составляет 60 CLB в высоту. Таким образом, общее количество центров распространения тактового сигнала в семействах UltraScale может достигать нескольких сотен, и эти центры равномерно распределены по всему кристаллу. Такая схема делает привлекательным использование GALS-архитектур для построения высокопроизводительных систем большого логического объема.

РЕАЛИЗАЦИИ ЦИФРОВОГО РЕКУРСИВНОГО ФИЛЬТРА С ИСПОЛЬЗОВАНИЕМ ПЛИС

*Научно-конструкторское бюро цифровой обработки сигналов
Южного федерального университета,
г. Таганрог,
dsp@sfedu.ru*

Во многих областях цифровой обработки сигналов приходится использовать цифровые фильтры. О структурах и способах их построения также имеется множество описаний [1-3]. Однако крайне мало статей о том, как именно организовать эти вычисления в ПЛИС. В данной работе рассмотрены два алгоритма реализации БИХ-фильтра с помощью ПЛИС.

Конкретная реализация цифрового фильтра в ПЛИС зависит от поставленной задачи. Если в ПЛИС требуется обрабатывать множество каналов с низкой частотой дискретизации, то необходимо сокращать количество аппаратных ресурсов, используемых для одного канала. Если же требуется высокая частота дискретизации, то количество используемых аппаратных ресурсов ПЛИС значительно возрастает. Отсюда вытекают две схемы реализации цифрового фильтра: с минимальными затратами аппаратных ресурсов ПЛИС или с максимально возможным быстродействием фильтра.

Рассмотрим разностные уравнения, описывающие БИХ-фильтр 2-го порядка, который часто используется для построения фильтров более высокого порядка путем их каскадного или параллельного соединения [2]:

$$\begin{aligned}w(n) &= x(n) - a_1 w(n-1) - a_2 w(n-2), \\y(n) &= b_0 w(n) + b_1 w(n-1) + b_2 w(n-2),\end{aligned}\tag{1}$$

где $x(n)$ – входной отсчет сигнала, $y(n)$ – выходной отсчет, a , b – коэффициенты фильтра, $w(n)$ – сигнал на выходе рекурсивной части фильтра.

Как видно, реализация этих уравнений с минимальным использованием аппаратных ресурсов ПЛИС требует одного умножителя, одного сумматора и временной памяти. Таким образом, уравнения (1) необходимо разделить на несколько более простых, где используется только одна операция умножения и одна сложения. Последовательность действий (операций) выполняемых по описанному выше алгоритму предполагается осуществлять следующим образом:

$$\begin{aligned}
 1) \quad & p_1 = -a_2 w_2, \\
 2) \quad & p_2 = -a_1 w_1, \\
 3) \quad & p_3 = b_2 w_2, \\
 4) \quad & p_4 = b_1 w_1, \quad w_0 = x + p_1, \\
 5) \quad & p_5 = 0, \quad w_0 = w_0 + p_2, \\
 6) \quad & p_6 = b_0 w_0, \quad y = p_3, \\
 7) \quad & y = y + p_4, \\
 8) \quad & w_2 = w_1, \quad w_1 = w_0, \\
 9) \quad & y = y + p_6,
 \end{aligned} \tag{2}$$

где $w_0 = w(n)$, $w_1 = w(n-1)$, $w_2 = w(n-2)$.

На рис. 1. показана схема реализации такого фильтра, который использует минимум аппаратных умножителей.

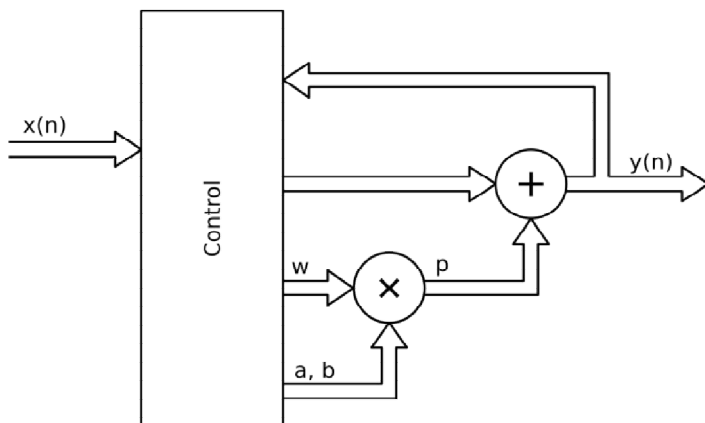


Рис. 1. Цифровой фильтр с одним умножителем

Принцип работы такой схемы заключается в следующем. На вход блока управления (*Control*) подается один отсчет входного сигнала $x(n)$. Затем этот блок последовательно во времени осуществляет подачу значений на умножитель и накапливающий сумматор в соответствии с (2). Умножитель содержит входные и выходные регистры, и результат умножения появляется только через 3 такта, поэтому на выполнение всех операций требуется 9 тактов и 1 такт на запись результата в выходной регистр. На 10-м такте на выходе сумматора формируется результат, который необходимо запомнить в выходном регистре.

В результате частота дискретизации сигнала F_d должна быть в 10 раз ниже частоты тактирования F_{clk} ПЛИС:

$$F_d \leq 0,1 F_{clk}. \quad (3)$$

Другой подход к проектированию цифрового фильтра показан на рис. 2.

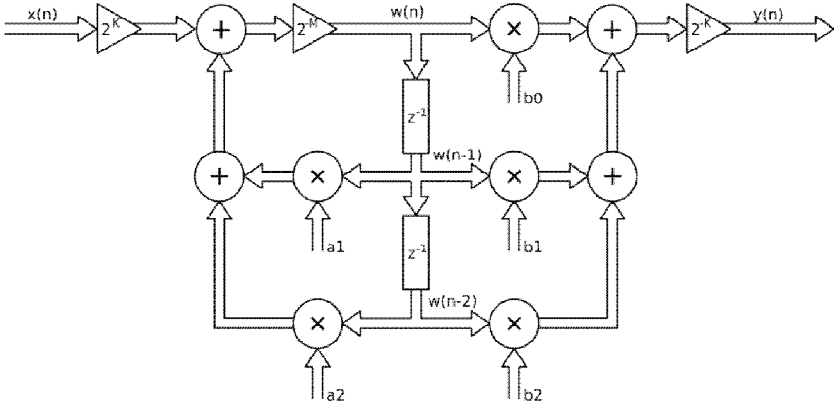


Рис. 2. Цифровой рекурсивный фильтр 2-го порядка

Здесь уже используются 5 аппаратных умножителей и 4 сумматора, то есть количество требуемых ресурсов заметно увеличивается. Такая реализация цифрового фильтра требует 2 такта на вычисление результирующего отсчета и 1 такт на его сохранение в выходной регистр. На первом такте выполняются все операции умножения и сложения задержанных сигналов. На втором такте выполняется сложение результата на выходе рекурсивной части со входным отсчетом, его умножение на коэффициент $b0$ и сложение с результатом на выходе нерекурсивной части фильтра.

Кроме того, при перемножении двух M -разрядных целых чисел на выходе умножителя получается число разрядности $2M$. В итоге, в эту схему были введены масштабные множители, чтобы не возникало переполнения разрядной сетки на выходе умножителей. Однако применение масштабных множителей сразу после умножителей подразумевает уменьшение точности представления чисел, поэтому целесообразнее сначала сложить числа, а потом выполнять масштабирование на 2^{-M} (отбрасывание младшей части числа).

Масштабные множители 2^K и 2^{-K} предназначены для предварительного усиления входного сигнала и его соответствующего ослабления на выходе фильтра. Если сумматоры имеют разрядность $2M$, то

входной сигнал необходимо усилить для согласования с масштабом сигнала на выходе умножителей. Аналогичная процедура выполняется и на выходе цифрового фильтра, только теперь согласование выполняется со входной частью модуля, который следует за фильтром.

В докладе изложены только схемы реализации цифровых фильтров, однако значения масштабных множителей разрядной сетки чисел и других параметров фильтра зависят от каждой конкретной реализации фильтра, используемой ПЛИС, заданных частот тактирования, дискретизации и других конкретных параметров. Применение одной из этих схем зависит от поставленной задачи и требует последующего точного описания фильтра на Verilog или VHDL и задания всех параметров фильтра.

1. Лайонс Р. Цифровая обработка сигналов: Второе издание. Пер. с англ. – М.: ООО «Бином-Пресс», 2006 г. – 656 с.
2. Оппенгейм А., Шафер Р. Цифровая обработка сигналов. Москва: Техносфера, 2006. – 856 с.
3. Цифровые фильтры и их применение: Пер. с англ./В. Каппелини, А. Дж. Константи́нидис, П. Эмилиани. – М.: Энергоатомиздат, 1983 – 360 с.
4. Угольников А.В., Цветков Ф.А. Реализация фильтра 10-го порядка на основе аппаратного модуля Xilinx Spartan-3AN Starter Kit // Материалы международной научной конференции «Системы и модели в информационном мире», ч. 3. Таганрог, 2009. С. 61–68.
5. Поляков А.К. Языки VHDL и Verilog в проектировании цифровой аппаратуры. – М.: СОЛОН-Пресс, 2003. – 320 с.
6. Бабак В. П., Корченко А. Г., Тимошенко Н. П., Филоренко С. Ф. VHDL: Справочное пособие по основам языка. – М.: Издательский дом «Додэка-XXI», 2008. – 224 с.
7. Altera Corporation. Cyclone II Device Handbook. [Электронный ресурс]. URL: http://www.altera.com/literature/hb/cyc2/cyc2_cii5v1.pdf. (дата обращения: 12.05.2014).

РАЗДЕЛ 4

Распределенные вычислительные и управляющие системы

*И.В. Бычков, Г.А. Опарин, А.Г. Феокистов,
В.Г. Богданова, А.А. Пашинин*

МУЛЬТИАГЕНТНОЕ УПРАВЛЕНИЕ РАСПРЕДЕЛЕННЫМИ ВЫЧИСЛЕНИЯМИ НА ОСНОВЕ СЕРВИС-ОРИЕНТИРОВАННОГО ПОДХОДА

*Институт динамики систем и теории управления СО РАН, г. Иркутск,
bychkov@icc.ru*

Введение

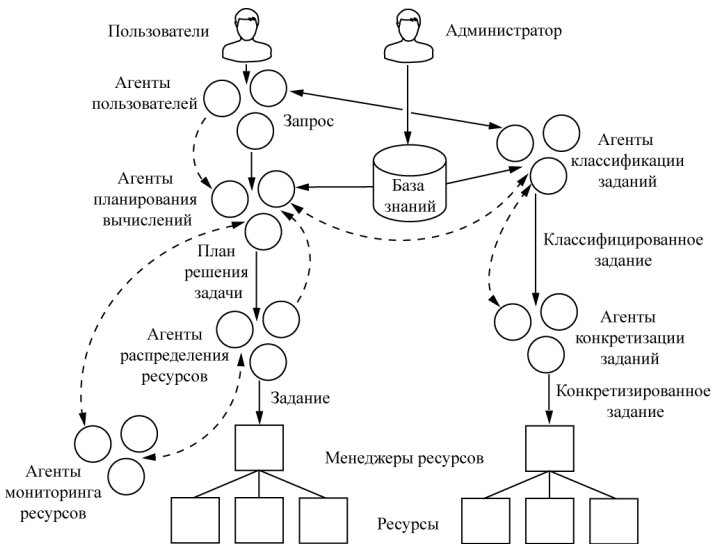
При организации распределенных вычислительных сред важной практически значимой проблемой является размещение в них проблемно-ориентированных программных комплексов, обеспечение удаленного высокоуровневого доступа пользователей-«предметников» к этим комплексам и управление ими в процессе вычислений. В настоящее время решение данной проблемы осуществляется, как правило, на основе концепции Service-Oriented Science, базирующейся на автоматизации интеграции распределенных разнородных ресурсов в виртуальную исполнительную среду и прозрачного использования этой среды путем представления ресурсов в виде сервисов. Известно применение научных сервисов в вычислительной химии, компьютерной алгебре, статистическом анализе временных рядов (статистическом прогнозировании), биоинформатике, геоинформатике и во многих других областях. Сервис-ориентированные среды обладают рядом свойств, существенно усложняющих процессы их организации, исследования и применения. В их числе: организационно-функциональная разнородность и динамичность; разнообразие спектра решаемых задач; наличие различных категорий пользователей, преследующих свои субъективные цели эксплуатации тех или иных ресурсов; применение в узлах среды различных средств управления вычислениями. Анализ мировых тенденций в области автоматизации решения прикладных задач в таких средах позволяет утверждать, что решение этих проблем непосредственно связано с интеллектуализацией сервис-ориентированного инструментария.

В докладе представлен подход к оформлению приложений, ориентированных на работу в распределенной вычислительной среде, в виде

удаленно доступных сервисов с помощью инструментальных средств, разрабатываемых в Институте динамики систем и теории управления СО РАН (ИДСТУ СО РАН).

Управление вычислениями

В рамках предлагаемого в докладе подхода управление вычислениями реализуется мультиагентной системой (МАС) с заданной организационной структурой, представленной на рисунке. Координация действий агентов реализуется с помощью общих правил группового поведения. Агенты функционируют в соответствии с заданными ролями, и для каждой роли определены свои правила поведения в виртуальном сообществе агентов. МАС включает агентов постановки задачи, планирования вычислений, мониторинга и распределения ресурсов, классификации, конкретизации и выполнения заданий. В разных виртуальных сообществах, возникающих в МАС, агенты могут координировать свои действия путем кооперации или соперничества.



Структура МАС

Администратор кластерной Grid-системы формирует классы заданий и выполняет виртуальную декомпозицию ресурсов, заключающуюся в назначении каждому ресурсу классов заданий, наиболее подходящих по своим характеристикам вычислительным возможностям этого ресурса. Сформированные классы заданий и результаты виртуальной декомпозиции ресурсов сохраняются в базе знаний. Агент постановки

задачи обеспечивает пользователю возможность формулировки задачи, определения требований к вычислительной системе, необходимых для ее решения, и управления процессом вычислений. Данный агент взаимодействует с пользователем Grid-системы в интерактивном режиме.

Агент планирования вычислений формирует в общем случае множество планов решения задачи. Подбор ресурсов для выполнения планов решения задачи и формирование заданий вычислительной системе осуществляют агенты распределения ресурсов. Агенты планирования вычислений и распределения ресурсов используют данные о состоянии этих ресурсов, собираемые агентами мониторинга ресурсов, и информацию о статусе уже запущенных на выполнение заданий, предоставляемую агентами выполнения заданий. Агенты классификации и конкретизации заданий, используя знания о классах заданий и результатах виртуальной декомпозиции ресурсов, осуществляют детальную настройку требований к вычислительной системе, содержащихся в заданиях, с целью обеспечения более эффективного планирования и распределения ресурсов планировщиками менеджеров ресурсов вычислительных кластеров (ВК) при обработке этих заданий.

Методы мультиагентного управления потоками заданий детально рассмотрены в работе [1]. При таком управлении время выполнения отдельных приложений может увеличиваться, поскольку при планировании не удастся учесть ряд важных особенностей структуры заданий и пользовательских предпочтений, касающихся ресурсов. В докладе представлены новые дополнительные средства планирования вычислений и распределения ресурсов на уровне приложений, позволяющие в определенной степени решать эту проблему.

Архитектура виртуального сообщества агентов

Дополнительные средства планирования вычислений и распределения ресурсов на уровне приложений представляют собой виртуальное сообщество (ВС) агентов, реализуемое для организации параллельного выполнения приложения пользователя в распределенной вычислительной среде. Основное назначение ВС агентов – обеспечение выбора наименее загруженного ресурса (кластера) для запуска на нем параллельного приложения, мониторинг его выполнения и передача результатов решения пользователю. ВС агентов включает пользовательского агента, агентов классификации и планирования, агента-менеджера и динамически меняющийся набор локальных агентов. Первые три агента осуществляют прием и классификацию запроса пользователя, построение плана вычислений. Сформированное в результате совместных действий агентов задание передается агенту-менеджеру, который, взаимодействуя с агентом мониторинга ресурсов, распределяет задание по локальным агентам ВК. Агент-менеджер отвечает также за выполнение следующих

функций: разбиение задачи на подзадачи, автоматический перезапуск задач с новыми параметрами (для определенного класса задач); управление, мониторинг выполнения задания пользователя. Локальные агенты отвечают за отправку заданий менеджеру ресурсов ВК, анализ текущего состояния ВК, передачу результатов агенту-менеджеру.

Для распределения задач по локальным агентам применяется известный протокол координации агентов, действующий по системе контрактов. Суть данного подхода состоит в том, что, получив спланированное задание пользователя, агент-менеджер рассылает предложения о выполнении данного задания на ВК своим агентам-подрядчикам, которые на основе данных о текущем состоянии кластера формируют ставку на выполнение данного задания. Впоследствии менеджер по ставкам выбирает наиболее подходящий ВК для проведения вычислений.

Особенностью представленного в докладе ВС является реализация агентов в виде сервисов. В состав инструментальных средств для реализации рассматриваемого подхода входят разработанная авторами библиотека классов и утилиты создания агентов на базе нейронных сетей, библиотека шаблонов сервисов на базе REST и SOAP архитектур, а также готовые сервисы на их основе. Кроме того в состав этих ИС входит описание хранения настроек сервисов в файлах XML. В качестве языка разработки выбран Java, для разработки soap-сервисов применены спецификации Java EE и сервер сервисов Apache TomEE.

Основными компонентами ВС являются агенты-менеджеры, которые и производят основную работу по распределению заданий, подбор наилучшей вычислительной среды под конкретную задачу, анализ состояний СУПЗ, перераспределение заданий, их декомпозицию. Данные агенты представлены в виде rest-сервисов. Для выполнения различных функций на вычислительных кластерах, зависящих от конфигурации ВК, они запускают soap-сервисы и должны иметь возможность их динамического подключения без перекомпиляции. Пользователи могут подключаться к любому из агентов-менеджеров через web-интерфейс сервиса, доступный с компьютеров и мобильных устройств, подключенных к сети Интернет, при условии наличия учетной записи на связанном с ним ВК.

Вычислительный эксперимент

При создании сервисов приложений в качестве распределенной вычислительной среды была использована разнородная кластерная Grid-система ИДСТУ СО РАН, включающая: однородный ВК «Blackford» с SMP-узлами, неоднородный ВК «Академик В.М. Матрозов» с SMP-узлами и узлом с графическими процессорами NVidia C2070 («Fermi»); неоднородный ВК «Tesla», в состав которого входит 4 четырехъядерных процессора и 8 GPU NVidia «Tesla» C1060.

В качестве первого примера был реализован сервис для решения задачи построения области устойчивости в пространстве двух выбранных параметров K и T регулятора замкнутой системы управления, описываемой дифференциальным уравнением $dx/dt = Ax$, где элементы матрицы A зависят от параметров K и T . В качестве второго примера реализован сервис для решения SAT-задач путем проведения многовариантных расчетов на основе декомпозиции исходной задачи.

В первом примере сервис приложения предназначен для реализации многовариантных расчетов с использованием статического планирования вычислений. Очевидно, что для задач такого вида преимущество рассмотренных выше средств мультиагентного управления по сравнению со стандартизированными менеджерами ресурсов незначительно. Во втором же примере сервис приложения предназначен для реализации многовариантных расчетов в режиме динамического планирования. В этом случае предлагаемые средства мультиагентного управления вычислениями позволяют существенно сократить время решения задачи по сравнению со стандартизированными менеджерами ресурсов за счет выбора наиболее оптимальных решателей и возможности миграции заданий по узлам вычислительной среды.

Заключение

Отметим отличительные особенности представленного подхода. Во-первых, в рамках данного подхода обеспечивается возможность применения средств разработки и выполнения сервисов приложений в разных режимах: использование веб-ориентированного пользовательского интерфейса Grid-сервисов и/или их спецификаций на языке WSDL; управление вычислениями как на уровне отдельных приложений, так и на уровне потоков заданий; распределение ресурсов, необходимых для выполнения операций сервисов, специальными агентами или стандартизированными менеджерами ресурсов, применение статического или динамического планирования вычислений. Во-вторых, создание ВС агентов, функционирующего на уровне приложений, позволяет представить в виде Grid-сервисов ряд системных функций этих приложений. В их числе: планирование вычислений, распределение и мониторинг ресурсов, определение статуса заданий, учет специфики проблемной области решения задачи в процессе управления вычислениями.

1. Бычков И.В., Опарин Г.А., Феоктистов А.Г., Кантер А.Н. Мультиагентный алгоритм распределения вычислительных ресурсов на основе экономического механизма регулирования их спроса и предложения // Вестник компьютерных и информационных технологий. 2014. № 1. С. 39-45.

GRIDMD: КОМПАКТНАЯ ПЕРЕНОСИМАЯ БИБЛИОТЕКА C++ ДЛЯ УПРАВЛЕНИЯ РАСПРЕДЕЛЕННЫМИ ВЫЧИСЛЕНИЯМИ

*Объединенный институт высоких температур РАН, г. Москва,
valuev@ihed.ras.ru*

Введение

В данной работе мы представляем текущее состояние открытой библиотеки GridMD [1,2], предназначенной для управления распределёнными вычислениями. Библиотека изначально создавалась для разработчиков молекулярно-динамических (МД) приложений. В настоящее время сфера применения библиотеки значительно расширена, и она служит универсальным средством для создания новых приложений, поддерживающих распределённые вычисления, а также для управления удалёнными задачами из компактного клиентского приложения средствами языка программирования C++. Библиотека GridMD – это программное средство, предназначенное для программистов-разработчиков, проектирующих приложения, использующие различные вычислительные ресурсы (в том числе удалённые) для исполнения отдельных вычислительных задач. В докладе будет представлен анализ разработанной библиотеки в сравнении с другими средствами управления расчетами, кратко описана архитектура и нестандартные возможности GridMD, а также описан опыт практического применения библиотеки для многомасштабных физических расчетов [3].

GridMD как система управления сценариями

Системы управления сценариями (workflow systems) [4-5] могут быть классифицированы как инфраструктурные и универсальные. Инфраструктурные проекты требуют поддержания определенной постоянно действующей среды (инфраструктуры), через которую осуществляется авторизация пользователей и постановка задач. В качестве такой базовой инфраструктуры может выступать Grid (проекты Unicore, Triana, gUSE), распределенная база данных (проект Taverna). Универсальные системы могут взаимодействовать с широким классом вычислителей. Существующие универсальные системы (Kepler, Pegasus), как правило, используют планировщик Condor как основное средство запуска заданий на разных классах вычислительных ресурсов.

Еще одним критерием классификации может служить то, для кого предназначена система: для конечного пользователя или разработчика промежуточного программного обеспечения. Большинство систем ориентировано на конечного пользователя, который непосредственно работает с определенным графическим интерфейсом и составляет в нем сценарий расчета. При этом специализированная настройка под конкретную задачу происходит на уровне отдельных элементов сценария (составных блоков, доступных к выбору в интерфейсе). Эти блоки настраиваются разработчиками промежуточного ПО так, чтобы отражать предметную область задачи (см., например, проект gUSE). Преимуществом такого подхода является универсальность интерфейса системы управления сценарием. Однако существует и недостаток: разработчик промежуточного ПО для предметной области не обладает достаточными средствами управления на уровне сценариев, и ему навязываются программные решения, определяемые высокоуровневым интерфейсом управления.

В отличие от большинства существующих систем и платформ управления распределенными сценариями библиотека GridMD является компактным универсальным клиентским средством, ориентированным на разработчика. Для своего функционирования она не требует установки и поддержки сложных инфраструктур, таких как Грид-системы, порталы, системы совместного пользования, однако может запускать вычислительные задачи на них от имени пользователя. Кроме того, GridMD может использоваться для разработки предметно-ориентированных расчетных платформ, требующих низкоуровневого (скрытого от конечного пользователя) управления сценариями вычислений. Пример такого использования обсуждается ниже.

Архитектура и функциональность

Внутри библиотеки GridMD выделяются два уровня виртуализации ресурсов, с которыми может взаимодействовать программа пользователя: «менеджер сценариев» и «менеджер заданий». В обычном режиме менеджер сценариев, обрабатывая граф исполнения, отправляет менеджеру заданий запросы на выполнение отдельных наборов команд и пересылку данных. При этом менеджер заданий имеет унифицированный интерфейс для различных способов доступа к удаленной системе, а также для различных способов выполнения, сформированных по общему образцу заданий, начиная от простейшего выполнения скрипта в командном интерпретаторе до постановки параллельной задачи в очередь с использованием различных систем очередей.

В функции менеджера заданий входит подготовка удаленной системы к выполнению сценария, копирование входных данных для каж-

дой отдельной задачи, запуск задачи с помощью выбранной системы очередей, контроль за ее выполнением и копирование выходных данных. Базовой подсистемой менеджера заданий является набор функций для выполнения удаленных команд и копирования файлов между локальной и удаленной системой. Разработчик может применять эти функции отдельно от оставшейся части библиотеки (без менеджера сценариев), получая, таким образом, высокоуровневый стандартизированный интерфейс для выполнения команд (Unix или Windows shell) на локальной или удаленной системе. Доступ к удаленной системе осуществляется с помощью встроенного протокола ssh. В настоящее время менеджер заданий поддерживает запуск задач на системах под управлением PBS(torque), SLURM, СУППЗ, Globus, а также непосредственно запуск и контроль процессов в ОС Unix и Windows.

Менеджер сценариев GridMD использует стандартный способ формулировки потока обработки (workflow), основанный на направленном ациклическом графе сценария. Основными элементами графа сценария являются узлы и связи. С узлами ассоциируются определенные действия программы, а со связями – данные (либо файлы данных), являющиеся их результатами. Входящие и исходящие связи представляют зависимости между узлами. Считается, что система может исполнить все действия узла, если известны результаты всех входящих в узел связей. В интерфейс разработчика входят такие функции как конструирование графа сценария (указание действий и данных), назначение критерия выбора ресурсов для исполнения конкретного узла, мониторинг состояния узлов в процессе исполнения, создание контрольных точек и перезапуск сценария.

Дополнительные возможности менеджера сценариев

Помимо стандартных средств работы с графом сценария, GridMD предоставляет разработчику ряд дополнительных возможностей, являющихся отличительными особенностями библиотеки. К ним относятся: использование процедур или участков кода клиентского приложения в качестве элементов сценария, динамическое управление графом сценария, алгоритмические шаблоны для автоматического создания элементов сценария.

Обычно системы управления сценариями запускают задачи, являющиеся какими-либо командами или сторонними приложениями. Библиотека GridMD позволяет сформулировать вычислительную задачу в виде процедуры, описанной на языке C++ и встроенной в управляющее клиентское приложение. Такая процедура может быть объявлена узлом графа сценария, т.е. получать управление и данные в порядке, определяемом сценарием вычислений. При этом, если та-

кая процедура выполняется удаленно, то на используемом вычислительном ресурсе должна быть доступна копия клиентского приложения, скомпилированная для соответствующей операционной системы. При запуске такой копии (в данном случае в качестве расчетного приложения) требуемая встроенная процедура будет найдена и исполнена автоматически. В случае, когда встроенная процедура запускается на стороне клиента, переноса клиентского приложения на удаленные вычислительные системы не требуется. В приложении GridMD могут определяться действия, модифицирующие граф исполнения задачи. Таким образом, граф сценария GridMD является динамическим. Для корректного восстановления графа сценария при перезапуске модифицирующие действия должны быть привязаны к событиям, соответствующим изменению состояния графа, при этом последовательность динамических модификаций графа может быть однозначно восстановлена. Возможность контроля событий графа обеспечивается интерфейсом разработчика GridMD.

Алгоритмические шаблоны, или “алгоритмические скелетоны” (algorithmic skeletons), – термин, который применяется для обозначения высокоуровневых моделей программирования для параллельных или распределенных вычислений [6-8]. Существует стандартная классификация алгоритмических шаблонов, наиболее употребительные среди которых – последовательность (pipeline), ветвление (fork), отображение (map) и др. Преимущество алгоритмического шаблона заключается в том, что для использования реализуемого им алгоритма необходимо только создать его наполнение, т.е. указать те конкретные процедуры, которые будут выполняться на определенных этапах алгоритма. Все служебные действия, связанные с передачей данных, взаимодействием между параллельно запущенными процессами и т.д., выполняются автоматически и скрыты от пользователя алгоритмического шаблона.

Пример практического использования

В докладе рассматривается существующий в настоящий момент опыт использования библиотеки GridMD в составе разработанных программных платформ для многомасштабного моделирования. За отчетный период открытый код GridMD использован компанией ООО «Кинтех Лаб.» для создания программного обеспечения в Российско-европейских проектах по моделированию нанокompозитов (CompNanoComp) и органических светодиодов (IM3OLED) [3]. Обе платформы реализованы на одной функциональной основе и представляют собой программные комплексы, объединяющие несколько расчетных программ и предназначенные для численного моделирования в конкретной предметной области. Библиотека GridMD используется в

обоих комплексах на низком уровне для управления сценариями расчетов. Под низким уровнем в данном случае понимается то, что конечный пользователь не имеет непосредственного доступа к элементам сценария GridMD (графу сценария и элементарным исполняемым заданиям), а составляет сценарий расчета из более крупных блоков. Сами расчеты могут производиться как на удаленных вычислительных ресурсах (кластерах), так и на локальной машине в зависимости от конфигурации расчетных приложений.

Накопленный опыт применения библиотеки сторонним разработчиком является очень важным. Он позволил уточнить типичные требования к возможностям библиотеки распределенных вычислений. Многие из требований к программному интерфейсу и к протоколам работы с удаленными вычислителями, сформулированные сторонним разработчиком, были учтены в новой версии GridMD. В частности, это относится к инкапсуляции протокола SSH внутрь библиотеки, созданию групп заданий, динамическому добавлению элементов сценария, реализации менеджера заданий для планировщика SLURM и другие. В целом, опыт использования GridMD показал, что библиотека является очень полезным средством разработчика. Она предъявляет минимальные требования к составу программного обеспечения на управляющей машине, обеспечивает требуемую функциональность по управлению заданиями минимальными средствами.

1. Morozov I.V., Valuev I.A. Automatic Distributed Workflow Generation with GridMD Library // Computer Physics Communications. 2011. V. 182. P. 2052–2058.
2. Программа в открытом доступе на <http://gridmd.sourceforge.net>
3. Integrated Multidisciplinary & Multiscale Modeling for Organic Light-Emitting Diodes: <http://www.im3oled.eu>
4. W.M.P. van der Aalst // The Journal of Circuits, Systems and Computers, V. 8, No. 1, P. 21 (1998);
5. Pytlinski, J., Skorwider, L., Benedyczak, K., et al // LNCS, V. 2658, P. 307 (2003)
6. Horacio González-Vélez and Mario Leyton "A survey of algorithmic skeleton frameworks: high-level structured parallel programming enablers" // Software: Practice and Experience 2010. V. 40. Issue 12. P. 1135.
7. eSkel project web site: <http://homepages.inf.ed.ac.uk/abenoit1/eSkel>.
8. Skandium project web site: <http://skandium.niclabs.cl.i>

ОБУЧЕНИЕ СИСТЕМ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ ДЛЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

*Объединенный институт проблем информатики НАН Беларуси, г. Минск,
viattchenin@mail.ru*

Введение

При неуклонно увеличивающемся объеме информации в процессе решения самых разнообразных задач роль распределенных вычислительных систем непрерывно возрастает. Распределенная обработка информации традиционно трактуется как проводимый на независимых, но связанных между собой вычислительных машинах, предназначенных для выполнения общих задач, комплекс операций с информацией. В свою очередь, возможность взаимодействия вычислительных систем при реализации распределенной обработки информации определяется как их способность к совместному использованию данных или к совместной работе с использованием стандартных интерфейсов.

Появление и широкое распространение распределенных вычислительных систем обусловлено, с одной стороны, снижением стоимости вычислительных средств и увеличением их производительности, что является следствием высоких темпов развития микроэлектроники, а с другой стороны – необходимостью более высокой производительности и эффективности вычислительных систем, обусловленной спецификой решаемых задач и сферами применения систем распределенной обработки информации.

Распределенные вычислительные системы, представляющие собой многомашинные вычислительные комплексы и компьютерные сети, являются одной из наиболее прогрессивных форм организации средств вычислительной техники. Именно в силу указанного обстоятельства значительно возрастает роль средств защиты таких систем от различных угроз.

В докладе обсуждаются основные подходы и методы, используемые при проектировании различных систем обнаружения вторжений, таких как системы обнаружения аномалий и системы обнаружения злоупотреблений, а также освещается вопрос о формировании обучающей выборки на основе базы KDD'99, традиционно используемой для обучения и тестирования систем обнаружения вторжений, для чего кратко рассматривается структура базы KDD'99 и определяются наиболее информативные признаки, а также объем формируемой для обучения выборки.

Подходы к проектированию систем обнаружения вторжений

Под системой обнаружения вторжений подразумевается система, осуществляющая сбор информации с множества системных и сетевых источников, анализирующая полученную информацию на предмет признаков вторжений [1]. Как уже отмечалось выше, системы обнаружения вторжений условно делятся на два типа: системы обнаружения аномалий и системы обнаружения злоупотреблений. В основе систем обнаружения аномалий лежит построение профиля нормального поведения субъекта, отклонение от которого трактуется как аномалия. Главным достоинством систем подобного рода является то, что с их помощью могут быть обнаружены не только известные, но и новые атаки. Эта особенность систем обнаружения аномалий является основополагающей при разработке перспективных систем обнаружения вторжений в условиях лавинообразного роста различных видов угроз.

В противоположность системам обнаружения аномальной активности, системы обнаружения злоупотреблений основаны на построении профилей атак, так что системы подобного рода не просто обнаруживают, но и распознают атаки. Вместе с тем, недостатком систем такого рода является как раз неспособность выявления атак нового типа, так что, исходя из достоинств и недостатков обоих подходов, наиболее перспективным представляется их комбинирование при разработке новых систем обнаружения вторжений.

При разработке как систем обнаружения аномалий, так и систем обнаружения злоупотреблений традиционно используются различного рода классификаторы, основанные преимущественно на методах мягких вычислений, включающих нейросетевые методы, методы нечеткой логики и генетические алгоритмы [2].

База данных KDD'99

Одной из главных проблем при сравнении различных подходов к обнаружению вторжений является то обстоятельство, что в различных исследованиях в качестве исходных данных используются различные наборы данных. Таким образом, в конце XX века возникла необходимость использования стандартизированного набора данных для обучения и тестирования различных систем с целью проведения их качественного сравнения.

В 1999 году была подготовлена база данных обработанного трафика проекта DARPA, получившая название KDD'99 [3] и состоящая из почти пяти миллионов записей о соединениях, из которых 20% представляют собой нормальный трафик, а остальные 80% несут информацию об атаках 22 типов, объединяемых в четыре класса: DOS – отказ в обслуживании, U2R – неавторизованное получение привилегий на дан-

ной системе, R2L – неавторизованный доступ к атакуемой системе с удаленной системы, Probe – различного рода зондирование атакуемой системы.

В качестве признакового пространства базы данных KDD'99 используется 41 параметр TCP-соединений, из которых 38 параметров представляют собой численные атрибуты, а 3 параметра – категориальные.

Формирование обучающей выборки на основе базы KDD'99

Так как большая часть параметров TCP-соединений является избыточной для проведения анализа соединений, различные исследователи использовали то или иное подмножество данных параметров. С этой целью зачастую используются методы отбора информативных признаков – к примеру, в [4] для этого используется метод главных компонентов, благодаря чему размерность признакового пространства была снижена с 41 до 10. С другой стороны, в работе [5] было предложено использовать следующее подмножество признаков, которые дают наиболее полное представление о состоянии узлов сети в данный момент:

- длительность соединения в секундах;
- используемый протокол;
- используемая служба прикладного уровня;
- количество отправленных байт;
- количество полученных байт;
- флаг TCP-соединения;
- количество соединений к текущему хосту за последние 2 с;
- количество соединений к текущей службе за последние 2 с;
- процент соединений к данной службе среди всех обращений;
- процент соединений к другим службам;
- количество соединений к удаленному хосту в течение последних 2 с;
- количество соединений к удаленной службе в течение последних 2 с;
- процент соединений к данной удаленной службе;
- процент соединений к другим удаленным службам;
- процент соединений к данному хосту при текущем номере порта источника;
- процент соединений к данной службе от разных хостов.

Данные параметры соединений являются наиболее значимыми из базы KDD'99. Следует также указать, что в [5] используется также временной штамп, не являющийся признаком, а служащий для идентификации записи в выборке.

На качество классификации существенно влияет также объем обучающей выборки: к примеру, в [5] в качестве выборки случайным образом взято 500 записей, характеризующих нормальный трафик, и по 500 записей для каждого из четырех видов атак; в работе [4] для обучения используется выборка объемом 4 911 записей. Вместе с тем, с увеличением объема обучающей выборки возрастает также скорость обучения системы. Для преодоления указанного недостатка была сформирована обучающая выборка общим объемом 494 записи, из которых 247 записей характеризуют различные виды атак, представленные в таблице.

Таким образом, остальные 247 записей являются наблюдениями за обычной работой сети, не подвергающейся атакам. Количественно соотношение записей атак в обучающей выборке соответствует соотношению их в полной базе.

Виды атак в обучающей выборке

№	Категория атаки	Вид атаки	Количество строк в обучающей выборке
1	U2R	perl	1
2	R2L	satan	22
3	U2R	rootkit	5
4	DOS	teardrop	11
5	R2L	pod	14
6	DOS	back	30
7	R2L	phf	1
8	Probe	portsweep	20
9	U2R	loadmodule	3
10	Probe	ipsweep	18
11	R2L	warezmaster	2
12	R2L	ftp_write	4
13	R2L	nmap	17
14	DOS	smurf	23
15	R2L	spy	1
16	R2L	warezclient	5
17	R2L	multihop	5
18	DOS	neptune	55
19	R2L	imap	2
20	DOS	land	1
21	U2R	buffer_overflow	3
22	R2L	guess_passwd	4

Заключение

Несмотря на то, что построенная указанным образом выборка составляет 0,01 процента от общего объема базы KDD'99, она может использоваться для обучения различных систем обнаружения вторжений с целью сравнения их эффективности. В частности, при обучении системы обнаружения вторжений, предложенной в [4], результат распознавания незначительным образом отличается от результата, полученного при обучении системы на выборке, содержащей 4 911 запись. Таким образом, предложенный метод формирования обучающей выборки может быть использован для сравнения эффективности систем обнаружения вторжений при моделировании атак на распределенные вычислительные системы.

1. Bace, R.G. Intrusion Detection / R.G. Bace. – Indianapolis: Macmillan Technical Publishing, 2000. – 349 p.
2. Azad, C. Data Mining in Intrusion Detection: A Comparative Study of Methods, Types and Data Sets / C. Azad, V.K. Jha // International Journal of Information Technology and Computer Science. – 2013. – Vol. 5, № 8. – P. 75-90.
3. KDD Cup'99 Competition [Electronic resource]. – MIT Lincoln Laboratory, 1999. – Mode of access: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99html>. – Date of access: 14.09.2013.
4. Husagic-Selman, A. Intrusion Detection System using Fuzzy Logic / A. Husagic-Selman // Southeast Europe Journal of Soft Computing. – 2013. – Vol. 2, № 1. – P. 14-20.
5. Кочурко, П.А. Обнаружение сетевых атак на компьютерные системы на основе нейросетевых технологий: дис. ... канд. техн. наук: 05.13.19 / П.А. Кочурко. – Брест, 2012. – 164 л.

А.Н. Голиков

АЛГОРИТМ ПРИБЛИЖЕНИЯ ФУНКЦИЙ НА ОСНОВЕ ПОЭКСТРЕМУМНОГО РАЗБИЕНИЯ ОТРЕЗКА ИНТЕРПОЛЯЦИИ С ЕСТЕСТВЕННЫМ ПАРАЛЛЕЛИЗМОМ

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
alex.golikov@mail.ru*

Введение. Необходимость обработки аналоговых и цифровых сигналов, поступающих на вход параллельной вычислительной системы, обуславливает необходимость приближения этих сигналов с целью их дальнейшей обработки, хранения, вычисления, в частности, скорости изменения уровня сигнала, обесшумливания и т.д. При этом требуется, чтобы используемые методы приближения обладали минимизирован-

ной абсолютной погрешностью, выполнялись за минимальное время и требовали по возможности наименьшего объёма памяти. Методы приближения функций востребованы также при решении дифференциальных уравнений, возникающих в задачах управления движением объектов, а также моделирующих нестационарные и нелинейные процессы, протекающие в узлах информационных систем, например при моделировании установления рабочего режима датчиков и микросхем обрабатывающих сигналов. При этом сигналы и временные зависимости искомым величин обрабатываются в виде приближений их аналитических представлений.

В настоящем докладе предлагается метод кусочно-многочленного приближения функций, производных и определённых интегралов на основе поэкстремумного разбиения отрезка интерполяции. Функция приближается интерполяционными многочленами Ньютона. В качестве значений приближаемой функции берутся значения дискретизированных сигналов или узловые значения приближаемого решения. Предлагаемый алгоритм нацелен на использование в многопроцессорных вычислительных системах, обладает естественным параллелизмом, под которым понимается наличие независимых последовательностей – нитей – вычислений, которые могут обрабатываться независимо друг от друга. Разбиение отрезка выполняется по известной из [1] схеме на основе сортировки, которая может быть воплощена на последовательных и параллельных вычислительных системах с логарифмической оценкой временной сложности.

Кусочно-многочленное приближение функций. Пусть дана функция

$$f : \mathbb{R} \supset [a, b] \rightarrow \mathbb{R}, f \in \mathcal{C}^{(N+1)}[a, b], \quad (1)$$

где $\mathcal{C}^{(N+1)}[a, b]$ – класс функций, непрерывных на $[a, b]$ вместе со своими производными до $(N + 1)$ -го порядка включительно.

На начальном шаге приближения отрезок $[a, b]$ разбивается упорядоченными по возрастанию экстремумами и нулями функции (1):

$$X^{(0)} = \{x_i^{(0)}\}_{i=0}^K. \quad (2)$$

Определение. Множество (2) назовём вершинами разбиения.

Все экстремумы и нули локализируются при помощи схемы на основе сортировки [1] с априори заданной верхней границей абсолютной погрешности, ограниченной снизу только погрешностью формата представления данных в вычислительной системе. Схема локализации использует сортировку, взаимно однозначно сопоставляющую индексы входного и выходного массивов; собственно экстремумы локализируются подпрограммой, использующей только операции сравнения и взятия абсолютной величины числа, поэтому схема не накапливает погрешность [1].

Конкретнее, по условию

$$|e[k+l] - e[k]| > \epsilon_{LOC}, \quad l = 1, 2, \dots, \tilde{n} - k, \quad (3)$$

проверяется, является ли элемент входного массива с индексом $e[k]$ локальным максимумом, то есть имеются ли в его ϵ_{LOC} -окрестности элементы, большие его. Аналогично, минимумы локализуются по условию

$$|e[k-l] - e[k]| > \epsilon_{LOC}, \quad l = 1, 2, \dots, \tilde{n}, \quad (4)$$

где e – массив, хранящий для каждого l -го элемента выходного массива его номер во входном, $\tilde{n}$ – размер массивов.

Нули идентифицируются как минимумы модуля значения функции.

Схема обладает естественным параллелизмом, так как все проверки условий (3) и (4) для каждого узлового значения функции могут быть выполнены независимо друг от друга.

На каждом подынтервале $[x_i^{(0)}, x_{i+1}^{(0)}]$, $i = \overline{0, K-1}$, с границами из (2) строится интерполяционный многочлен Ньютона с приведением при помощи параллельной матричной схемы вычисления коэффициентов многочлена по его корням к каноническому виду [2]:

$$P_{i,n}(x) = \sum_{j=0}^n a_{i,j} x^j. \quad (5)$$

Шаги используемой матричной схемы инвариантны относительно степени полинома, схема оперирует только целочисленными данными, в следствие чего не копится погрешность округления [1].

Степень многочлена (3) выбирается алгоритмически минимальной, при которой выполняется условие

$$|f(x) - P_{i,n}(x)| \leq \epsilon, \quad (6)$$

где ϵ – априори заданная граница абсолютной погрешности приближения; левая часть (4) вычисляется в проверочных точках, расположенных с равным шагом меньшим шага интерполяции.

Конечные разности i -го порядка многочлена (4) вычисляются по формуле

$$\Delta^m y_i = \sum_{k=0}^m (-1)^{m-k} C_m^k f(y_i + kh), \quad (7)$$

где биномиальные коэффициенты C_m^k считаются априори известными и помещёнными в память компьютера, а произведения $C_m^k f(y_i + kh)$ и собственно сумма (7) вычисляются по схеме сдваивания (см. рисунок).

Время массово параллельного вычисления разностей (7) без учёта времени вычисления значений функции (1) и времени на диспетчеризацию на $\frac{n^2}{2}$ процессорах составит [3]

$$T\left(\frac{n^2}{2}\right) = t_y + \lceil \log_2(m+1) \rceil t_c,$$

где t_y и t_c – время одного умножения и сложения соответственно.

Если в априори заданных ограничениях сверху на степень не удастся удовлетворить условию (4), то на текущем подынтервале степень n многочлена (3) выбирается либо равной значению, при котором абсо-

лутная величина старших конечных разностей всё ещё меньше, чем при предыдущем n , либо равной предельно максимальному значению в данных ограничениях на степень.

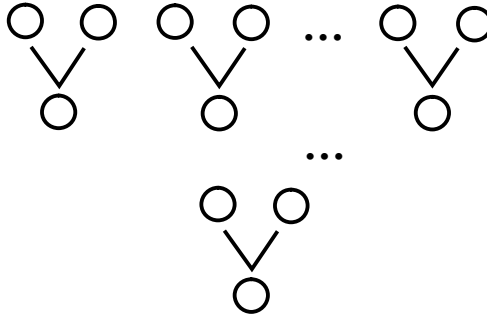


Схема сдвигания

Затем, аналогично локализации экстремумов приближаемой функции, на основе сортировки локализуются экстремумы первой производной от построенного многочлена, при этом производная $\frac{d}{dx} P_{i,n}(x)$ вычисляется по схеме Горнера.

Вершины разбиения (2) – экстремумы и нули функции, вычисленные на предыдущем шаге, – пополняются экстремумами производной, после чего на полученном новом разбиении $\mathcal{X}^{(1)} \supset \mathcal{X}^{(0)}$ построение интерполянта повторяется. Если на некотором подынтервале не удаётся удовлетворить условию (6), то вершины разбиения пополняются экстремумами второй производной от полинома (3), тогда построение приближающего многочлена повторяется на новом разбиении $\mathcal{X}^{(2)} \supset \mathcal{X}^{(1)} \supset \mathcal{X}^{(0)}$, и так далее.

Пополнение разбиений $\mathcal{X}^{(k)}$ нулями очередной $(k+1)$ -й производной продолжается либо до достижения значения $k = n - 4$ – наибольшего, при котором последовательность значений $\frac{d^{k+1}}{dx^{k+1}} P_{i,n}(x)$ ещё будет содержать экстремум в строгом смысле, либо до выполнения условия (6) на всех подынтервалах. В последнем случае многочлен (5) считается построенным, и его коэффициенты заносятся в память компьютера, после чего приближение функции (1), а также её производной и первообразной сведётся к вычислению по схеме Горнера многочлена (5), его производной и первообразной соответственно.

Замечание 1. Сходимость предложенного приближения имеет место в случае достаточной гладкости функции (1), при этом отрезок $[a, b]$ может быть заменён произвольным компактом в $\mathbb{R}$.

Замечание 2. При построении схемы предполагается, что ограничения сверху на степень многочлена соответствуют частоте дискретизации датчиков.

Замечание 3. Описанное преобразование интерполянта к каноническому виду позволяет, сохранив в памяти один лишь набор коэффициентов $a_{i,j}$ из (3), приближать функцию, её производные, первообразную и определённые интегралы по произвольному отрезку $[c, d] \subset [a, b]$, используя схему Горнера, или другую распараллеливаемую схему вычисления полинома. Временная сложность максимально параллельной формы преобразования приближающего многочлена к виду (3) за счёт использования матричной схемы из [1] имеет логарифмическую оценку.

Изложенная методика позволяет на практике снизить абсолютную погрешность приближения функций с быстро растущими производными, например функции Франке, а также снизить погрешность приближения производных и определённых интегралов при снижении объёма памяти, требуемого для сохранения коэффициентов (3).

Независимые вычислительные нити – потоки – имеют место при разбиении отрезка интерполяции, за счёт параллельности локализации экстремумов, при построении приближающего многочлена на каждом из параллельно обрабатываемых отрезков, а также при вычислении значения приближающего многочлена в точке. Таким образом, выполнение этих независимых вычислительных потоков на многопроцессорных системах позволит значительно снизить временную сложность обработки дискретизированного сигнала, а также снизить временную сложность и абсолютную погрешность приближённого решения дифференциальных уравнений при моделировании и управлении подвижными объектами.

1. Ромм Я.Е. Локализация и устойчивое вычисление нулей многочлена на основе сортировки // Кибернетика и системный анализ. Киев. 2007. № 1. С. 165–182.
2. Ромм Я.Е., Голиков А.Н. Кусочно-полиномиальная схема вычисления функций и определённых интегралов с повышенной точностью // Известия ЮФУ. Технические науки. Тематический выпуск «Методы и средства адаптивного управления в электроэнергетике». 2011. № 2. С. 38–44.
3. Аксайская Л.Н. Разработка и исследование параллельных схем цифровой обработки сигналов на основе минимизации временной сложности вычисления функций // Автореферат диссертации на соискание ученой степени кандидата технических наук. – Таганрог: ТТИ ЮФУ, – 2008.

ФОРМАЛИЗАЦИЯ ЗАДАЧИ ФОРМИРОВАНИЯ КОНФИГУРАЦИИ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ С УЧЕТОМ ПАРАМЕТРА ЗАГРУЗКИ ПРОЦЕССОРНЫХ УСТРОЙСТВ

*Инженерно-технологическая академия ЮФУ, г. Таганрог,
НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
g.v.gorelova@gmail.com, anna_klimenko@mail.ru*

Введение

В настоящее время одним из перспективных направлений в области исследования и разработки информационно-управляющих систем (ИУС) являются распределенные ИУС с децентрализованным управлением. Идея ИУС с децентрализованным управлением, или диспетчированием, кратко может быть изложена следующим образом [1]: основу ИУС составляют равноправные и взаимозаменяемые процессорные устройства (ПУ) под управлением унифицированного программного обеспечения. Если ИУС выполняется некоторая задача управления (ЗУ), и происходит сбой на одном или нескольких ПУ, такой принцип организации управления ИУС позволяет восстановить (перезапустить или продолжить) выполнение ЗУ, что является базовым отличием такой ИУС от ИУС с централизованным управлением, где в случае сбоя на центральном диспетчере система становится полностью неработоспособной. Процесс перезапуска (или продолжения выполнения) ЗУ на функционирующих ПУ в контексте данной работы будем называть реконфигурацией ИУС с децентрализованным диспетчированием. Соответственно, получение конфигураций – решение задачи формирования конфигурации ИУС. Целью данного доклада является анализ известных формальных постановок задачи формирования конфигурации и влияния на формализацию задачи добавления нецелочисленного параметра – степени загрузки ПУ.

Анализ известных постановок задачи и формализация задачи формирования конфигурации с учетом параметра загрузки ПУ

В самом общем случае, говоря о задаче формирования конфигурации, речь идет о распределении подзадач ЗУ по функционирующим и резервным (если такие имеются) ПУ [2,3].

Общие модели задачи [2,3] имеют следующие общие особенности:

- в качестве входных данных указывается ациклический граф, состоящий из подзадач, связанных информационно и (или) по управлению;
- исходное множество ПУ, из которых формируется архитектура ВС, может быть однородным или неоднородным, но количество ПУ различных типов не ограничено. Иными словами, может быть сформирована ВС из любого набора ПУ;
- в качестве исходных данных также указывается время, к которому должны быть завершены вычисления;
- предполагается, что один ПУ может в один момент времени решать только одну задачу, и каждая задача, имеющая входящие дуги, может быть взята на решение только тогда, когда завершено решение задач-предшественниц;
- решением задачи является система ПУ и составленное расписание решения ЗУ, представленной графом.

В качестве минимизируемой целевой функции, как правило, выбирается суммарная стоимость получаемой системы [2]. Также в работе [4] целевой функцией принято отклонение значений загруженности ПУ от среднего значения.

Задачу в описанной постановке следует отнести к задачам невыпуклой целочисленной условной оптимизации. Невыпуклость обеспечивается множественностью локальных экстремумов; целочисленность, в свою очередь – конечностью множества возможных перестановок параметров модели. Также следует отметить, что задачу реконфигурации и восстановления в выделенных постановках относят к классу NP-сложных, для которых в настоящее время не найден метод решения полиномиальной сложности.

Тем не менее, приведенные здесь формальные модели задачи имеют общий характер и не учитывают степени загруженности ПУ, которая является важным показателем для систем реального времени и в особенности для систем, сбои которых критичны. Учет величины загруженности каждого ПУ и введение в качестве дополнительного параметра ресурса производительности ПУ позволяет ослабить одно весьма существенное ограничение, а именно – одновременное выполнение ПУ только одной задачи. Также можно предположить, что введение параметра загруженности ПУ позволит получать расписания, удовлетворяющие поставленным ограничениям, на наборах ПУ меньшей стоимости.

Исходными данными для задачи формирования конфигурации будут следующие:

1. ЗУ, представленная ациклическим информационным графом $G = (N, X)$, где N – множество номеров вершин, X – трудоемкости подзадач в элементарных операциях.
2. Множество ПУ $M = \{<j, p_j, l_j, s_j>\}$, где j – номер ПУ, p_j – производительность, оп/с, l_j – показатель загруженности ПУ в момент начала решения задачи реконфигурации, s_j – стоимость ПУ. Здесь следует отметить, что в момент запуска режима реконфигурации функционирующий ПУ может быть занят решением другой ЗУ и иметь загруженность $l_j > 0$.
3. Директивное время завершения решения ЗУ $T_d > 0$.
4. Максимально допустимая степень загруженности ПУ после перераспределения подзадач ПУ $L_{max} < 1$.

Рассмотрим более подробно показатель загруженности ПУ и его влияние на формальную постановку задачи формирования конфигурации.

В том случае, когда заданная графом ЗУ должна быть решена к определенному моменту времени, для каждой из задач могут быть определены ранние и поздние сроки ее завершения. Обозначим через t_{ij} момент начала решения i -й задачи j -м ПУ, τ_{ij1} – ранний срок завершения решения i -й задачи j -м ПУ, τ_{ij2} – поздний срок завершения решения i -й задачи j -м ПУ.

Загруженность ПУ δ_{ij} при решении i -й задачи j -м ПУ при условии решения задачи не позже позднего срока будет минимальной в том случае, если время решения задачи займет $\tau_{ij2} - t_{ij}$ и, соответственно, будет максимальной при решении j -й задачи в случае $\tau_{ij1} - t_{ij}$.

Для $0 \leq \delta_{ij} \leq L_{max}$ будет справедливо выражение

$$\tau_{ij1} - t_{ij} \leq \frac{x_i}{\delta_{ij} p_j} \leq \tau_{ij2} - t_{ij}. \quad (1)$$

Формула (1) означает, что при условии выполнения подзадачи ЗУ не позднее указанного срока, степень загруженности j -го ПУ может варьироваться. Если учитывать требование к выравниванию нагрузки на ПУ, то помимо (1) добавляется ограничение на переменные δ_{ij} следующего вида:

$$\sum_{i,j} \delta_{ij} \leq L_{max} \quad (2)$$

для любого зафиксированного j и всех i -х задач, решаемых одновременно на j -м ПУ.

Обозначим через Δ_j суммарную загруженность ПУ.

$$\Delta_j = \sum_i \delta_{ij} . \quad (3)$$

Тогда условие выравнивания загруженности ПУ будет иметь вид

$$\Delta_1 = \Delta_2 = \dots = \Delta_m \leq L_{max} . \quad (4)$$

Также через M' обозначим результирующее множество ПУ, на которое будет производиться распределение подзадач, $M' \subseteq M$.

Следует отметить, что показатель загруженности ПУ $\delta_j \in Q$ и, не являясь целочисленным, переводит задачу формирования конфигурации ИУС в класс невыпуклых частично-целочисленных задач условной оптимизации.

Расписание, являющееся одним из результатов решения формализуемой задачи, может быть представлено несколькими способами. Например, в [5] предлагается представлять расписание в виде матрицы $m \times n$, где на пересечении строк и столбцов указываются начальные моменты решения подзадач. Также расписание может быть представлено в виде множества векторов, каждый из которых соответствует j -му ПУ и элементами которых являются начала выполнения i -х задач. Будем представлять расписание работ двумерной матрицей следующего вида:

$$R = \begin{bmatrix} t_{ij} & -1 & t_{kl} \\ \dots & \dots & \dots \\ \dots & \dots & -1 \end{bmatrix}, \quad (5)$$

где каждый элемент

$$r_{ij} = \begin{cases} t_{ij}, & \text{в случае назначения задачи,} \\ -1. & \end{cases}$$

Таким образом, учитывая сказанное выше, сформулируем задачу восстановления и реконфигурации следующим образом.

Для исходных данных, перечисленных в п.п.1-4, необходимо решить задачу формирования конфигурации таким образом, что

$$S = \min_{M', R, \Delta_j} \sum_j s_j$$

при следующих ограничениях:

- 1) $\Delta_1 = \Delta_2 = \dots = \Delta_m \leq L_{max}$;
- 2) $\forall j : 0 \leq \Delta_j \leq L_{max}$;
- 3) $\forall i, j : \max(t_{ij} + \frac{\delta_{ij} p_j}{x_i}) \leq T_d$;
- 4) $M' \subseteq M$;

5) $t_{ij} \geq 0$.

Отметим, что параметр загрузки ПУ имеет непрерывный характер на интервале $[0;1)$, что позволяет построить модель задачи, не привязанную к определенному типу ПУ.

Выводы

Учет степени загрузки ПУ является важным параметром при формировании структуры ИУС и планирования вычислений.

Введение переменной, определяющей степень загрузки ПУ и имеющей непрерывный характер, переводит задачу формирования конфигурации ИУС из класса задач целочисленного нелинейного программирования в класс задач условной оптимизации частично-целочисленного нелинейного программирования. Так как общих методов решения подобных задач в настоящее время не существует, подобная формулировка задачи реконфигурации ставит вопросы о поиске и разработке новых методов ее решения, тем самым определяя направление дальнейших исследований.

1. Каляев И.А., Мельник Э.В. Децентрализованные системы компьютерного управления. Ростов н/Д: Издательство ЮНЦ РАН, 2011. 196с.
2. Барский А.Б. Параллельные процессы в вычислительных системах: планирование и организация. – М.: Радио и связь, 1990
3. Костенко В.А. Возможности генетических алгоритмов для решения задач синтеза архитектур и планирования параллельных вычислений// Труды Третьей Международной научной конференции “Дискретные модели в теории управляющих систем” (Красновидово’98) М.: Диалог МГУ, 1998. - С.53-58.
4. Мельник Э.В., Горелова Г.В. Об эффекте выравнивания вычислительной нагрузки процессорных устройств в высоконадежных РИУС, планирование исследований // Мехатроника, автоматизация, управление. – 2012. – № 11. – С. 29-35.
5. Conway R.W., Maxwell W.L., Theory of Scheduling, Addison Wesley Publishing Company, Reading, Massachusetts, 1967.

ПОСТРОЕНИЕ ПРОСТЕЙШЕЙ ГРИД-СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ GLOBUS TOOLKIT

Южно-Российский региональный центр информатизации ЮФУ,
г. Ростов-на-Дону,
dolv@sfedu.ru

Структура и задачи грид

Создание грид-сети означает объединение территориально распределенных вычислительных ресурсов. Под ресурсами могут подразумеваться вычислительные узлы и/или узлы хранения и передачи данных, собственно данные, прикладное программное обеспечение. Нас интересуют непосредственно вычислительные ресурсы, которые предоставляют задаче пользователя процессорные мощности. Под вычислительными ресурсами могут пониматься как кластеры, так и многоядерные рабочие станции. При всем разнообразии архитектур любая вычислительная система может рассматриваться как потенциальный ресурс грид-сети [1].

Обычно грид-система (рис. 1) включает в себя следующие основные инфраструктурные компоненты [2]:

- совокупность компьютеров с установленными на них интерфейсами пользователя;
- совокупность Ресурсных центров, включающих в себя вычислительные ресурсы;
- ресурсы хранения данных;
- набор базовых грид-сервисов.

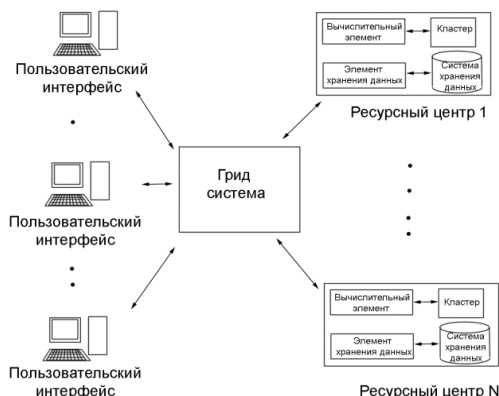


Рис. 1. Схема грид-системы

Интерфейс пользователя предоставляет доступ пользователя к ресурсам грида. С помощью интерфейса пользователь может выполнять следующие задачи:

- осуществляет запуск заданий на выполнение;
- пересылает данные с одного ресурса хранения данных на другой;
- контролирует процесс выполнения задания;
- получает результат выполнения задания.

Структура пакета GlobusToolkit

Пакет Globus Toolkit, который состоит из нескольких модулей, предназначен для построения виртуальной среды распределенных вычислений [3]. Каждый модуль имеет стандартный интерфейс, используемый высокоуровневыми компонентами, и реализован для различных систем. Все вместе модули позволяют создавать виртуальную машину Globus (рис. 2).

Все модули могут быть разделены на следующие группы:

- модули безопасности,
- модули управления данными,
- модули управления процессами,
- общий Runtime.

Globus Toolkit позволяет сохранить полную автономность всех вычислительных систем с локальным администрированием и с управлением всех заданий и локальных, и удаленных пользователей локальными диспетчерскими системами.

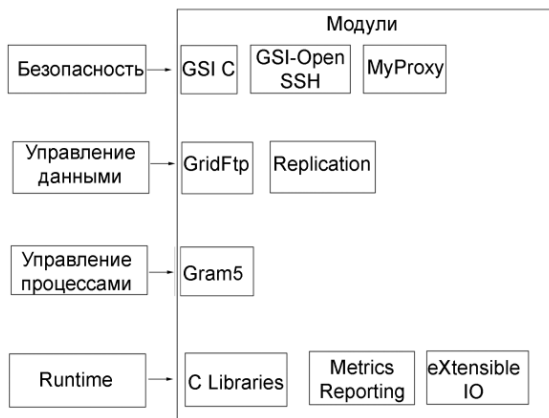


Рис.2. Архитектура Globus Toolkit 5.0.3

Построение грид

Для начала сформулируем задачу. Имеются три удаленных кластера, на каждом из которых работает независимая система аутентификации пользователей и диспетчерская система управления заданиями. В идеале пользователи каждого из них должны иметь возможность запускать свои задания на любом вычислительном ресурсе. Для решения поставленной задачи необходимо решить три подзадачи:

- разработать систему авторизации пользователей;
- привести в соответствие программное обеспечение для обеспечения бинарной совместимости программ;
- предоставить интерфейс взаимодействия с вычислительными ресурсами.

Предлагаемая вниманию система объединила вычислительные ресурсы департамента математики ЮФУ(ЮГИНФО; факультет математики, механики и компьютерных наук; Институт механики и прикладной математики).

Интерфейс пользователя грид-сети

Для начала сформулируем последовательность действий пользователя для выполнения задания на удаленном кластере:

- 1) получение сертификата;
- 2) передача на удаленный кластер исполнимой программы и необходимых данных;
- 3) запуск программы на удаленном кластере;
- 4) получение результатов с удаленного кластера.

Очевидно, что для реализации этой последовательности действий требуется некоторый набор команд. Кроме того, необходимы средства для контроля состояния удаленных кластеров и состояния выполняющихся заданий. Для выполнения всех перечисленных функций было написано 9 специальных команд, образующих интерфейс пользователя грид-сети.

```
gridinit SERVER_NAME
```

Здесь `SERVER_NAME` — имя удаленного грид-сервера. Возможные значения сегодня

```
mmcs      — для сервера МММКН;
```

```
niimpm    — для сервера НИИМиПМ.
```

Сертификат действителен в течение 12 часов.

Задание запускается командой

```
gridrun JOB PROG SERVER QUE NODES PROC CORES
```

`Tlim`, где

```
JOB      — имя PBS скрипта, под которым будет запущено задание;
```

```
PROG     — имя файла с исполнимой программой;
```

SERVER – имя грид-сервера (mmcs или niimp);
 QUE – имя очереди;
 NODES – число узлов (default 1);
 PROC – число процессов на каждом узле (default 1), если значение этого параметра равно 0, то программа считается не mpi-ная и вызывается как обычная однопроцессорная программа;
 CORES – число нитей на каждом процессе (default 1);
 TLim – лимит времени для задачи (default 336 Hour).

В тех случаях, когда запускающий скрипт имеет нестандартный вид, например в командной строке нужно передать программе аргументы, то его можно подготовить на локальном сервере с помощью текстового редактора. Тогда для запуска задания на удаленном кластере следует использовать команду

`gridqsub JOB PROG SERVER`, где

JOB – имя PBS скрипта, который был подготовлен заранее;
 PROG – имя файла с исполнимой программой;
 SERVER – имя грид-сервера (mmcs или niimp).

После завершения задания результаты остаются на удаленном сервере. В дальнейшем, независимо от того, успешно или нет завершилось задание, все содержимое рабочего каталога на удаленном сервере может быть переписано в рабочий каталог локального сервера командой

`gridget SERVER`, где

SERVER – имя грид-сервера.

Поскольку пользователь лишен права интерактивного входа на удаленные сервера, то для контроля состояния удаленных кластеров нужны специальные команды:

`gridpestat SERVER`, показывает состояние вычислительных узлов,

SERVER – имя грид-сервера;

`gridqstat SERVER [OPTION]`, показывает состояние очередей,

SERVER – имя грид-сервера;

OPTION – опции команды qstat.

Очевидно, что необходимы также команды для контроля выполнения задания. Для этой цели создана команда для просмотра содержимого рабочего каталога на удаленном сервере. Она позволяет убедиться, что создаются какие-то выходные файлы и в них пишется какая-то информация:

`gridls SERVER`,

SERVER – имя грид-сервера.

Как и команда **gridget**, она обязательно должна запускаться из исходного рабочего каталога на главном сервере.

Следующая команда позволяет просматривать содержимое текстовых файлов в рабочем каталоге на удаленном сервере:

```
gridcat SERVER FILE,
SERVER  – имя грид-сервера,
FILE    – имя файла в рабочем каталоге на удаленном сервере.
```

Команда обязательно должна запускаться из исходного рабочего каталога главного сервера.

Следующая команда позволяет просматривать содержимое каталога \$USER. Это может служить напоминанием о том, какие задания еще не завершены: либо не закончен расчет, либо данные еще не перемещены на локальный сервер.

```
gridlsa SERVER,
SERVER  – имя грид-сервера.
```

Наконец, если мониторинг выполнения задания показал, что с заданием что-то не в порядке, то его можно снять командой.

```
gridqdel SERVER NUMBER_TASK,
SERVER  – имя грид-сервера;
NUMBER_TASK – номер задания на удаленном сервере.
```

Вместо команды **gridinit** можно использовать команду `clfree SERVER_NAME`,

которая получит сертификат от удаленного сервера и покажет количество свободных узлов на удаленном кластере.

Заключение

Globus Toolkit предоставляет пользователям полноценное управление над распределёнными ресурсами. Globus хорошо подходит для конструирования своей собственной грид-системы на основе сервисов, включенных в пакет. С помощью Globus Toolkit можно легко создать необходимую среду выполнения вычислительных заданий. Это может быть как глобальный грид, объединяющий несколько вычислительных ресурсов, сильно удаленных географически, так и небольшая грид-система объединяющая ресурсы в рамках одного учреждения.

1. Ian Foster and Carl Kesselman (eds), "The Grid, Blueprint for a New computing Infrastructure", Morgan Kaufmann Publishers, Inc., 1998.
2. Ian Foster and C. Kesselman (eds), "The Grid 2: Blueprint for a New Computing Infrastructure", Morgan Kaufmann Publishers, 2004.
3. Проект Globus: <http://www.globus.org>.

**ЗАДАЧА СИСТЕМАТИЗАЦИИ НАУЧНО-ВЫЧИСЛИТЕЛЬНЫХ
СЕРВИСОВ РАСПРЕДЕЛЕННОЙ СРЕДЫ***

*Тверской государственный технический университет,
common@tstu.tver.ru*

При разработке проектных решений, инженерном моделировании процессов различной природы, конструировании исследуемых областей, автоматизации вычислительных задач и т.д. все большее применение находят компьютерные инфраструктуры, основанные на концепции облачных вычислений. Согласно [1], облачные вычисления – технология распределенных вычислений, в которой компьютерные ресурсы и мощности предоставляются пользователю как интернет-сервис.

Как правило, распределенные среды ориентированы на поддержку исследований в областях, объединенных тематическими научными направлениями. Например, распределенная среда для решения задач вычислительной химии, разработанная ИПХФ РАН [2], включает в себя сервисы для моделирования молекулярных структур и их динамики, исследования поведения и энергетики сложных химических реакций, наномоделирования и т.д.; распределенная математическая среда Wolfram Math World [3] направлена на поддержку различных математических исследований; сеть распределенных вычислений World Community Grid (IBM) поддерживает научные проекты в области медицины и т.д.

Для обеспечения доступа пользователей к сервисам и пакетам, содержащимся в распределенных средах, разработан ряд тематических информационных каталогов, таких как «Mathtree» – древовидный каталог математических интернет-ресурсов, разработанный институтом систем информатики им. А.П.Ершова СО РАН [4]; каталог биологических сервисов [5]; каталог различных сервисов [6] и т.д. При разработке структуры каталогов учитываются стандартные подходы к составлению каталогов, предложенные Декларацией о международных принципах каталогизации, принципы которой распространяются на область применения, объекты, атрибуты, связи, задачи и функции каталога. Также применяется опыт составления рубрикаторов, представляющих собой иерархическую классификацию областей знания, принятую для систематизации всего потока научно-технической информации.

Несмотря на обширно представленное направление каталогизации информации, обеспечение поиска данных, являющееся одним из самых проблемных разделов систематизации информации, в большинстве случаев осуществ-

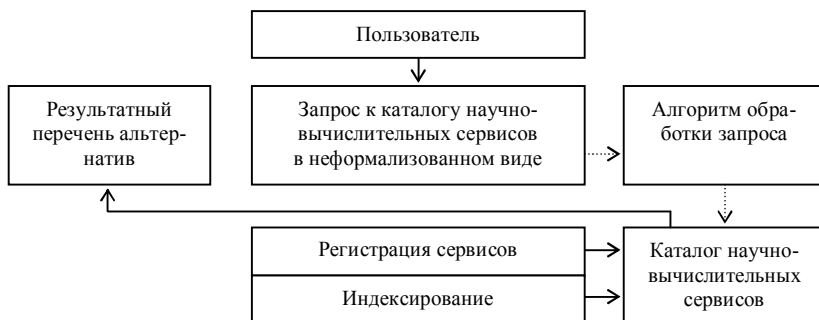
* Работа выполнена при поддержке РФФИ (13-07-00077, 13-07-00342).

ляется на основе строго сформулированного запроса, в результате выполнения которого пользователю представляется список искомых альтернатив.

Как показывает практика, зачастую запрос к системе невозможно четко сформулировать, ключевые слова могут не совпадать с заранее заданным перечнем термов в семантическом описании сервиса. Невозможно также однозначно указать критерии, на которых базируется поиск информации. Для выбора решения из представленного списка разделов, относящихся к определенной предметной области, необходимо самостоятельно выбирать раздел, к которому относится решаемая задача, что зачастую невозможно. Для того чтобы избежать ограничения доступа к полному объему информации, содержащейся в каталогах научно-вычислительных сервисов распределенных сред, целесообразно представлять запрос к системе в качественном виде, кроме того, такая информация лучше формализуется с помощью нечетких множеств.

Постановка задачи систематизации научно-вычислительных сервисов

Пусть имеется распределенная среда, содержащая набор научно-вычислительных сервисов, применение которых связано с решением научных и прикладных задач. Пользователь формирует запрос к системе с формулировкой подлежащей к решению задачи. Система обращается к каталогизированному набору сервисов с целью сформировать ранжированный перечень возможных альтернатив (см. рисунок). При этом предполагается, что число шагов отбора сервисов фиксированное.



Постановка задачи систематизации научно-вычислительных сервисов распределенной среды

Задача состоит в разработке алгоритма функционирования системы, позволяющего формировать результатный перечень научно-вычислительных сервисов, максимально удовлетворяющий запросу за фиксированное число шагов принятия решения.

Алгоритм систематизации научно-вычислительных сервисов распределенной среды

Пусть множество $X = \{x_j\}$ – каталог сервисов, содержащий сервисы $\{x_1, x_2, \dots, x_n\}$.

Шаг 1. Автор, включающий в каталог информацию, регистрирует сервис с помощью каталогизационной карты. Обозначим множество характеризующих атрибутов как $A = \{a_1(x_j), a_2(x_j), \dots, a_k(x_j), j = 1, \dots, n\}$.

Например, минимальный набор атрибутов при описании математического сервиса будет выглядеть следующим образом: название математического сервиса; описание математического сервиса; круг задач, для которых используется данный математический сервис; входные данные; выходные данные; форматы ввода и вывода данных; описание вычислительной процедуры; относительная вычислительная сложность задачи; относительная алгоритмическая сложность задачи; язык программирования, на котором написано приложение; скорость и время работы математического сервиса; переносимость ресурса; надежность; стоимость; корректность используемого алгоритма; необходимость дополнения/модификации сервиса; дата создания, изменения сервиса. В качестве дополнения можно указывать, когда и в каких разработках использовался данный математический сервис, а также успешность его применения.

Шаг 2. Следующим этапом описания сервиса является индексирование, необходимое для обеспечения быстрого прямого поиска информации по совпадению слов в запросе и в аннотации. Семантическое описание сервисов осуществляется с помощью ключевых слов, или термов. Набор семантических единиц возможно формировать, применяя алгоритмы деления текста на отдельные слова, при этом в качестве исходного семантического описания объекта принимаем аннотацию сервиса.

Обозначим $S = \{s_1(x_j), s_2(x_j), \dots, s_m(x_j), j = 1, \dots, n\}$ – множество семантического описание сервиса, характеризующееся семантическими единицами.

Шаг 3. Пользовательский запрос к системе, формулируемый в неформализованном виде можно представить в следующем виде:

$$L = L_1 \cup L_2 \cup L_3 \cup L_4,$$

где L_1 – множество термов из поисковой строки, L_2 – дополнительное множество слов-синонимов, схожих по значению и близких по смыслу слов; L_3 – дополнительное множество ассоциативных слов и словосочетаний; L_4 – множество переведенных ключевых слов в зависимости от настроек подключения словарей (при формировании множеств используются словари автоматической обработки текста).

Обозначим множество семантического описания, формируемое на основе пользовательского запроса как $L = \{l_1(s_j), l_2(s_j), \dots, l_q(s_j), j = 1 \dots m\}$.

Шаг 4. При выборе наиболее подходящего для решения поставленной задачи сервиса необходимо использовать заданное множество параметров $p_j \in P$, оценивающих содержание сервисов и их функциональность. Пользователь устанавливает приоритетные требования к сервисам $P = \{p_1(a_j), p_2(a_j), \dots, p_r(a_j), j = 1, \dots, k\}$.

Шаг 5. Для простоты обозначим пересечение нечетких множеств A и P как $Y = A \cap P$, пересечение множеств S и L – как $H = S \cap L$.

Для построения функции принадлежности μ_Y множества Y каждому $y_i(x_j)$ поставим в соответствие число $\mu_{y_i}(x_j)$, где $i=1, \dots, k, j=1, \dots, n$, $0 \leq \mu_{y_i}(x_j) \leq 1$. Аналогичным образом, для построения функции принадлежности μ_X множества X каждому $h_i(x_j)$ поставим в соответствие число $\mu_{h_i}(x_j)$, где $i=1, \dots, l, j=1, \dots, n$, $0 \leq \mu_{h_i}(x_j) \leq 1$. При $\mu_{y_i}(x_j) = 0$ и $\mu_{h_i}(x_j) = 0$ – по данному описанию сервис точно не подходит, при $\mu_{y_i}(x_j) = 1$ и $\mu_{h_i}(x_j) = 1$ – сервис точно подходит, при $0 < \mu_{y_i}(x_j) < 1$ и $0 < \mu_{h_i}(x_j) < 1$ – промежуточные варианты.

В результате получаем системы значений

$$\begin{cases} \mu_{y_1}(x_1), \mu_{y_2}(x_1), \dots, \mu_{y_k}(x_1) \\ \mu_{y_2}(x_2), \mu_{y_2}(x_2), \dots, \mu_{y_k}(x_2) \\ \dots \\ \mu_{y_k}(x_n), \mu_{y_k}(x_n), \dots, \mu_{y_k}(x_n) \end{cases}$$

и

$$\begin{cases} \mu_{h_1}(x_1), \mu_{h_2}(x_1), \dots, \mu_{h_l}(x_1) \\ \mu_{h_2}(x_2), \mu_{h_2}(x_2), \dots, \mu_{h_l}(x_2) \\ \dots \\ \mu_{h_l}(x_n), \mu_{h_l}(x_n), \dots, \mu_{h_l}(x_n). \end{cases}$$

Будем считать, что решение – это сформированное в процессе функционирования многошаговой системы поиска альтернатив множество сервисов D , максимально отвечающее поставленным в запросе условиям.

Тогда положим

$$\begin{aligned} \mu_Y(x_1) &= \min\{\mu_{y_1}(x_1), \dots, \mu_{y_k}(x_1)\} \\ \mu_Y(x_2) &= \min\{\mu_{y_1}(x_2), \dots, \mu_{y_k}(x_2)\} \\ &\dots \\ \mu_Y(x_n) &= \min\{\mu_{y_k}(x_n), \dots, \mu_{y_k}(x_k)\} \end{aligned}$$

и

$$\begin{aligned} \mu_H(x_1) &= \min\{\mu_{h_1}(x_1), \dots, \mu_{h_l}(x_1)\} \\ \mu_H(x_2) &= \min\{\mu_{h_1}(x_2), \dots, \mu_{h_l}(x_2)\} \\ &\dots \\ \mu_H(x_n) &= \min\{\mu_{h_l}(x_n), \dots, \mu_{h_l}(x_l)\}. \end{aligned}$$

Следуя формуле Беллмана-Заде будем представлять решение как слияние целей и ограничений. Тогда результатом решения поставленной задачи является множество $D = Y \cap H$, таким образом, цель совпадает с решением. Итак, для поиска подходящего научно-вычислительного сервиса из множества D выбирается тот сервис, у которого функция принадлежности будет наибольшей: $\mu_D(x) = \min\{\mu_Y(x), \mu_H(x)\}$.

Тогда оптимальное решение сводится к отысканию

$$x^* = \arg \max_x \mu_D(x).$$

Заключение

В данной работе приведено обоснование целесообразности использования основных принципов теории нечетких систем для обеспечения поиска научно-вычислительных сервисов в каталоге распределенной среды, осуществлена постановка задачи, описаны основные шаги алгоритма функционирования каталога сервисов при решении научных и прикладных задач. Очевидно, что описанная выше процедура отыскания x^* легко распараллеливается, задача решается перебором. При построении данных систем целесообразно использовать соответствующие сервисы, расположенные в распределенных средах.

1. Суперкомпьютерные технологии в науке, образовании и промышленности / Под редакцией: академика В.А. Садовниченко, академика Г.И. Савина, чл.-корр. РАН В.В. Воеводина. – М.: Издательство Московского университета, 2012. – 232 с.
2. Волохов В.М., Варламов Д.А., Пивушков А.В., Покатович Г.А., Сурков Н.Ф. Технологии ГРИД в вычислительной химии // Вычислительные методы и программирование. Т.11, 2010. С.42-29.
3. Проект Wolfram Mathworld <http://mathworld.wolfram.com> (дата обращения: 25.04.2014).
4. Каталог математических интернет ресурсов <http://www.mathtree.ru> (дата обращения: 25.04.2014)
5. Проект Biocatalogue <http://www.biocatalogue.org/> (дата обращения: 25.04.2014)
6. Проект Programmable web <http://www.programmableweb.com/> (дата обращения: 25.04.2014)
7. Беллман Р., Заде Л. Принятие решений в расплывчатых условиях. Мир, М.: 1976. 46 с.
8. Палюх Б.В., Егерев И.А. Многошаговая система поиска альтернатив в информационном каталоге // Программные продукты и системы. 2013. №3. С.291-295.
9. Иванов В.К., Виноградова Н.В. Эвристический алгоритм фильтрации и семантического ранжирования результатов поиска документов // Вестник Тверского государственного университета. Серия: Прикладная математика. 2013. № 3 (30). С. 97-106.

РАСПРЕДЕЛЕННОЕ УПРАВЛЕНИЕ БОЛЬШОЙ ГРУППОЙ РОБОТОВ ПРИ РЕШЕНИИ СТРОЕВОЙ ЗАДАЧИ*

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог
donat.ivanov@gmail.com*

Пространственное расположение мобильных роботов в группе в соответствии с требуемой формой называют строем (*англ. formation*), а задачу формирования строя заданной формы в группе мобильных роботов называют строевой задачей (*англ. formation task*).

Решение строевой задачи занимает важное значение в групповой робототехнике, так как применение мультиробототехнических комплексов требует формирования и поддержания заданных пространственных построений (строев, формаций) между роботами группы во многих практических приложениях, таких как видеомониторинг территории [1,2]; мониторинг акватории [3,4]; картографирование [5]; разведка [6]; поиск радиоактивных объектов [7]; формирование мобильных коммуникационных сетей [8]; формирование мобильных фазированных антенных решеток [9] и др.

В рамках данной работы под «большой группой роботов» будем понимать такую группу, для которой выполняется хотя бы одно из условий:

- радиус действия телекоммуникационного оборудования хотя бы одного робота меньше, чем размеры группы;
- объем данных, передаваемых роботами группы в общий канал связи, превышает возможности отдельного робота по обработке полученной информации;
- объем вычислений, необходимых для решения строевой задачи для всех роботов группы, превышает вычислительные возможности бортовых систем отдельных роботов.

При таком подходе группа с заданной численностью может быть и «большой» и «не большой» в зависимости от решаемых задач, применяемых алгоритмов управления и линейных размеров используемого строя.

Краткий обзор подходов к решению строевой задачи в группах роботов

Известны различные подходы к решению строевой задачи в мультиробототехнических комплексах, которые можно разделить на группы:

- метод потенциальных полей (Potential Field Approach) [10–12];

* Работа выполнена при финансовой поддержке РФФИ (проект №14-08-0176а)

- бионический (биомиметический) подход [13–16];
- подход по принципу «ведущий–ведомый» (leader–follower) [17–19];
- подход по принципу «виртуальная структура/виртуальный лидер» (virtual structure/virtual leader) [20];
- подход на основе теории игр [21];
- метод окружностей [22] и др.

Многие из этих подходов ориентированы на управление небольшими группами роботов, а те, которые можно применять и для управления большими группами роботов, позволяют формировать строй только некоторой определенной формы. Таким образом, актуальным является решение строевой задачи в большой группе роботов, позволяющее формировать строй необходимой формы.

Предлагаемый метод распределенного управления большой группой роботов при решении строевой задачи

Предлагаемый метод базируется на принципах роевого взаимодействия, изложенных [23], а также предполагает, что формируемый строй в большой группе роботов имеет периодическую структуру (что обусловлено отсутствием полного графа коммуникации между роботами в группе). Тогда можно выделить повторяющиеся элементы, которые будем называть «паттернами». К примеру паттерном может быть квадрат, треугольник, шестиугольник или прямоугольник. Паттерном может служить и более сложная фигура.

Каждой вершине паттерна присваивается идентификатор (например буква латинского алфавита) и предъявляются следующие требования: известна ориентация либо на сторону света, либо на другую вершину паттерна, известны ориентация и дистанция до вершины с таким же идентификатором (буквой) другого паттерна.

Требования к роботу: наличие уникального номера-идентификатора; связь с соседними роботами; способность определять сторону света и азимут на соседа; способность измерять дистанцию до соседних роботов; мобильность; способность избегать столкновений с другими роботами группы за счет небольших отклонений от курса.

Алгоритм решения строевой задачи в большой группе роботов:

- 1) каждой вершине паттерна назначается идентификатор-буква (см. рис. 1, а);
- 2) распределение ролей роботов группы в определенной пропорции (например: поровну А, Б и В в каждой локальной области) с помощью алгоритма распределения задач, описанного в [24] (см. рис. 1, б);
- 3) в каждой локальной области роботы с ролью «А» позиционируются в соответствии с требованиями к вершине «А» паттерна, роботы с ролью «Б» ищут себе не занятую пару «А» и выстраиваются относительно нее на нужную дистанцию и азимут, далее роботы с

ролью «В» и ищут себе незанятую пару из «А» и «Б» и так далее до тех пор, пока не будет сформировано необходимое количество паттернов. Некоторые роботы могут остаться без места в паттерне и будут выполнять резервную роль (см. рис. 1, в).

- 4) происходит перемещение паттернов друг относительно друга с предотвращением столкновений:
 - а. если два паттерна пересекаются друг относительно друга, то они «вытесняют» друг друга из себя. Для этого происходит поиск геометрических центров паттернов и для каждого из паттернов строится вектор с началом в геометрическом центре и направлением, противоположным направлению на геометрический центр второго паттерна. Движение паттернов по этим векторам позволяет разъединить паттерны;
 - б. если два паттерна не пересекаются друг с другом, то они сближаются так, чтобы расстояния между A_i и A_j ($i < j$) были на требуемой дистанции, указанной в описании формируемого строя (см. рис. 1, з).

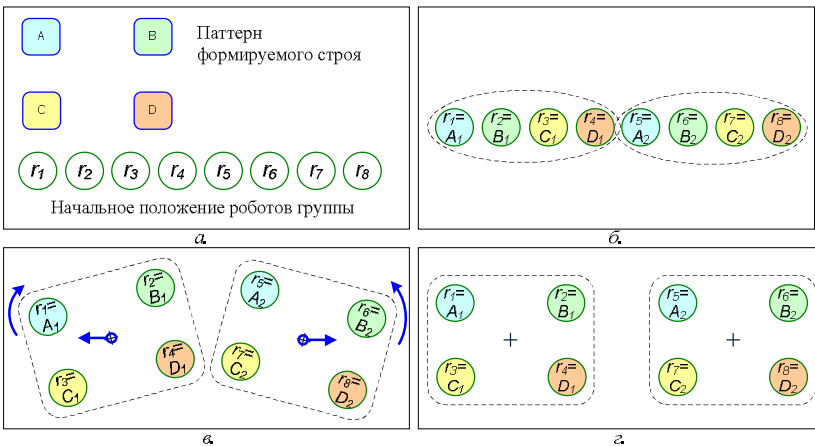


Рис. 1. Решение строевой задачи на примере группы из восьми роботов и прямоугольного паттерна

Закключение

Предложенный подход позволяет решать строевую задачу для формирования строя периодической структуры в больших группах мобильных роботов, при этом отличается низкой вычислительной сложностью и может находить применение даже в условиях ограниченных коммуникаций между роботами группы.

1. Иванов Д. Формирование строя группой беспилотных летательных аппаратов при решении задач мониторинга // Известия ЮФУ. Технические науки. 2012. Вып. 4. С. 219–224.
2. Casbeer D.W. et al. Forest fire monitoring with multiple small UAVs // Proc. 2005, Am. Control Conf. 2005. IEEE, 2005. P. 3530–3535.
3. Туфанов И.Е., Щербатюк А.Ф. Разработка алгоритмов группового поведения АНПА в задаче обследования локальных неоднородностей морской среды // Управление большими системами. 2012. Вып. 36. С. 262–284.
4. Киселев Л.В. et al. Ситуационное управление группировкой автономных подводных роботов на основе генетических алгоритмов // Подводные исследования и робототехника. 2009. Вып. 2. С. 34–43.
5. Кучерский Р.В., Манько С.В. Алгоритмы локальной навигации и картографии для бортовой системы управления автономного мобильного робота // Известия ЮФУ. Технические науки. 2012. Вып. 123, № 3. С. 13–22.
6. Прокопович Г.А., В.А.Сычев. Моделирование поискового поведения стаи роботов // 4-я Всероссийская мультikonференция по проблемам. Таганрог: Изд-во ТТИ ЮФУ, 2011. С. 286–289.
7. Chadwick R.A. Operating Multiple Semi-Autonomous Robots: Monitoring, Responding, Detecting // Proc. Hum. Factors Ergon. Soc. Annu. Meet. SAGE Publications, 2006. Vol. 50, № 3. P. 329–333.
8. Градецкий В.Г., Ермолов И.Л., Князьков М.М. Построение подвижных коммуникационных сетей на базе наземных автономных мобильных роботов // Мехатроника. Автоматизация. Управление. 2011. Вып. 11.
9. Tonetti S. et al. Distributed control of antenna array with formation of uavs // World Congr. 2011. Vol. 18, № 1. P. 7848–7853.
10. Рыжова Т.П. Управление коллективом мобильных роботов // Экстремальная робототехника. Труды международной научно-технической конференции. 2011. С. 281–287.
11. Барбашова Т., Кирильченко А.А., Колганов М.А. Некоторые аспекты использования метода потенциалов при управлении мобильными роботами. Москва, 2004.
12. Sutanty D., Kernbach S., Levi P. Multi-Robot searching algorithm using Lévy flight and artificial potential field // IEEE Int. Work. Safety, Secur. Rescue Robot. 2010.
13. Balch T., Arkin R.C. Behavior-based formation control for multirobot teams // IEEE Trans. Robot. Autom. 1998. P. 926–939.
14. Reynolds C.W. Flocks, herds and schools: A distributed behavioral model // ACM SIGGRAPH Comput. Graph. 1987. Vol. 21. P. 25–34.
15. Funge J., Tu X., Terzopoulos D. Cognitive modeling: knowledge, reasoning and planning for intelligent characters // Proc. 26th Annu. Conf. Comput. Graph. Interact. Tech. 1999. P. 29–38.
16. Brogan D.C., Hodgins J.K. Group behaviors for systems with significant dynamics // Proc. 1995 IEEE/RSJ Int. Conf. Intell. Robot. Syst. Hum. Robot Interact. Coop. Robot. 1995. Vol. 3.
17. Wang P.K.C. Navigation strategies for multiple autonomous mobile robots moving in formation // J. Robot. Syst. 1991. Vol. 8, № 2. P. 177–195.

18. Desai J.P., Ostrowski J., Kumar V. Controlling formations of multiple mobile robots // Proceedings. 1998 IEEE Int. Conf. Robot. Autom. (Cat. No.98CH36146). 1998. Vol. 4.
19. Mesbahi M., Hadaegh F.Y. Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching // Proc. 1999 IEEE Int. Conf. Control Appl. (Cat. No.99CH36328). 1999. Vol. 2.
20. Lewis M.A., Tan K.-H. High Precision Formation Control of Mobile Robots Using Virtual Structures. // Auton. Robot. 1997. Vol. 4. P. 387–403.
21. Erdoğan M.E., Innocenti M., Pollini L. Obstacle Avoidance for a Game Theoretically Controlled Formation of Unmanned Vehicles // 18th IFAC. 2011.
22. Иванов Д.Я. Метод окружностей для формирования строя в группе мобильных роботов // IX Ежегодная научная конференция студентов и аспирантов базовых кафедр Южного научного центра РАН тезисы докладов конференции (г. Ростов-на-Дону, 11-24 апр. 2013 г.). 2013. С. 115–116.
23. Иванов Д.Я. Алгоритмы организации групповых действий малоразмерных беспилотных летательных аппаратов при различных схемах применения // 4-я Всероссийская мультikonференция по проблемам управления. 2011. Р. 284–286.
24. Иванов Д.Я. Модель роевого взаимодействия в больших группах роботов в условиях ограниченных коммуникаций // V Ежегодная научная конференция студентов и аспирантов базовых кафедр Южного научного центра РАН. Тезисы докладов (8-27 апреля 2009 г., г.Ростов-на-Дону). 2009. С. 131–133.

А.И. Каляев

МЕТОДЫ ПОВЫШЕНИЯ НАДЕЖНОСТИ СИСТЕМ ПЕРЕДАЧИ ДАННЫХ В КОРПОРАТИВНЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
anatoly@kalyaev.net*

На сегодняшний день программно-конфигурируемые сети (ПКС) используются для организации высокопроизводительных вычислительных сетей, рассчитанных на передачу больших объемов данных. [1] В то же время зачастую ПКС используется для построения виртуальных сетей для каждой из задач передачи данных, действующих в ПКС одновременно [2-7].

При создании систем управления (СУ) (и СУ ПКС в том числе) на основе современной элементной базы наилучших показателей по соотношению показателей быстродействие – надежность – стоимость можно добиться, если использовать сетевую организацию и скользящее резервирование. Основное достоинство сетевой организации с точки зрения обеспечения надежности – широкие возможности реконфигурации, ос-

новное достоинство скользящего резервирования – универсальность резерва. При этом необходимо учитывать, что возможны различные варианты повышения надежности: наряду со стандартным вариантом скользящего резервирования на уровне контроллеров возможен и другой вариант – на уровне имеющихся в контроллерах резервов ресурсов.

Для этого предлагается ввести виртуализацию на уровне системы управления ПКС: разбить ПУ системы на несколько независимых подмножеств ПУ, каждое из которых будет ответственно за свою виртуальную сеть. Однако в данном случае возникает вопрос: каким образом организовать подмножества? Если выделить для этого специальный узел, – он станет узким местом системы, и надежность работы системы в целом упадет. Для того чтобы сохранить надежность работы системы на высоком уровне, предлагается воспользоваться принципами коллективного взаимодействия, а точнее мультиагентным подходом, предполагающим одновременную работу множества независимых друг от друга сущностей – агентов, которые в процессе взаимодействия приходят к определенным договоренностям, которые в свою очередь определяют в итоге эмергентное поведение системы. На сегодняшний день принципы мультиагентного взаимодействия успешно применяются при решении множества различных задач [8-10].

Для этого на каждом ПУ системы будет запущен специальный программный модуль – агент мультиагентной системы, который будет контролировать свой ПУ. Тогда станет возможным создание независимых сообществ, каждое из которых сможет осуществлять управление своей виртуальной сетью, поддерживать необходимую отказоустойчивость и надежность, а вместе с тем и распределять нагрузку управления на множестве ПУ, из которых будет состоять.

Для более удобного восприятия информации введем определение «задача» – (сокращение от «задача управления») задача управления набором коммутаторов, составляющих определенную виртуальную сеть, которые обслуживаются единым управляющим модулем. То есть для каждой виртуальной сети в СУ ПКС существует своя задача. Задача должна иметь срок давности, так как её редактирование в рамках децентрализованной ПКС является сложной задачей (по аналогии с временем жизни правила в коммутаторе ПКС).

Особенностью предлагаемой схемы организации СУ ПКС является возможность оперативно добавлять новые задачи, а следовательно и новые виртуальные сети. При этом для решения каждой полученной системой задачи должна быть сформирована виртуальная организация множества агентов – сообщество. В данной работе предлагается под сообществом в дальнейшем понимать коллектив агентов, собравшихся вместе для решения одной и той же определенной задачи. [6] При этом состав сообщества может изменяться в процессе решения задачи. Отказы оборудования ПУ, изменение их производительности, приводящие к

увеличению времени отклика системы управления выше требуемого, могут вызвать изменение состава сообщества.

В ходе работы автором были предложены принципы функционирования СУ ПКС с мультиагентным диспетчером, которые подробнее изложены в докладе.

1. Коровин Я.С., Каляев А.И. Методы и средства повышения производительности систем передачи данных в корпоративных сетях нефтегазовых компаний / Нефтяное хозяйство. 2013. № 9. С. 96-100. (URL: <http://elibrary.ru/item.asp?id=20282874>)
2. Коровин Я.С., Хисамутдинов М.В. Метод получения незашумленного изображения на основе обработки видеопоследовательности/ Журнал "Компьютерная оптика", ISSN 0134-2452, Россия, Самара //Том 38 N1, январь - март 2014, с. 112-118. (URL: <http://elibrary.ru/item.asp?id=21348534>)
3. Коровин Я.С., Хисамутдинов М.В., Ткаченко М.Г. Прогнозирование состояния нефтепромысловых объектов с применением технологий эволюционных алгоритмов и искусственных нейронных сетей/ Научно-технический и производственный журнал "Нефтяное хозяйство", ISSN 0028-2448// №12, 2013, с.128-133. (URL: <http://elibrary.ru/item.asp?id=20931355>)
4. Я.С. Коровин, С.В. Кононов, М.Г. Ткаченко. Оперативная диагностика состояния нефтепромыслового оборудования на основе технологий интеллектуальной обработки данных. /Научно-технический и производственный журнал "Нефтяное хозяйство", ISSN 0028-2448// №9, 2012, с.116-118. (URL: <http://elibrary.ru/item.asp?id=17940546>)
5. Коровин Я.С., Хисамутдинов М.В. Метод определения разновысотности тепловыделяющих сборок в задаче телевизионного контроля тепловыделяющих сборок водо-водяного энергетического реактора. Вестник компьютерных и информационных технологий. 2012. № 12 (102). С. 3-7 (URL: <http://elibrary.ru/item.asp?id=18922521>)
6. Коровин Я.С. Система поддержки принятия решений по контролю состояния установок электроцентробежных насосов на основе нейронной сети. Нефтяное хозяйство. 2007. № 1. С. 80-83 (URL: <http://elibrary.ru/item.asp?id=9466490>)
7. Коровин Я.С., Хисамутдинов М.В. Система распознавания номеров тепловыделяющих сборок водо-водяного энергетического реактора с применением интеллектуального анализа данных. Вестник компьютерных и информационных технологий. 2013. № 1 (103). С. 3-8. (URL: <http://elibrary.ru/item.asp?id=18921214>)
8. Каляев А.И. Метод и алгоритмы адаптивной организации распределенных вычислений в децентрализованной GRID / Вестник компьютерных и информационных технологий. 2012. № 4. С. 28-33. (URL: <http://elibrary.ru/item.asp?id=17681405>)
9. Каляев А.И. Метод и алгоритмы адаптивной организации распределенных вычислений в децентрализованной GRID (ОКОНЧАНИЕ) / Вестник компьютерных и информационных технологий. 2012. № 5. С. 37-41. (URL: <http://elibrary.ru/item.asp?id=17730358>)
10. Каляев А.И. Децентрализованная организация диспетчера GRID на базе сообществ агентов / Известия Южного федерального университета. Технические науки. 2011. Т. 121. № 8. С. 230-238. (URL: <http://elibrary.ru/item.asp?id=16562793>)

ЭКСПЕРИМЕНТАЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ ИМИТАЦИИ ОТЖИГА ПРИ РЕШЕНИИ ЗАДАЧИ ФОРМИРОВАНИЯ КОНФИГУРАЦИИ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
anna_klimenko@mail.ru*

Введение

Информационно-управляющие системы (ИУС) в настоящее время широко используются практически повсеместно, начиная от бытовых приборов и заканчивая сложными технологическими комплексами атомной энергетики и нефте- газодобывающей промышленности [1].

На этапе проектирования должна быть решена задача размещения подзадач задачи управления (ЗУ) по процессорным устройствам (ПУ) и получены варианты допустимых конфигураций ИУС.

Формирование конфигурации ИУС

Неформальное представление задачи формирования конфигурации ИУС в самом общем виде выглядит следующим образом: необходимо осуществить распределение частично упорядоченного множества подзадач ЗУ по имеющимся в наличии ПУ таким образом, чтобы ЗУ была решена в установленные сроки. Аналогичные задачи рассмотрены в литературе неоднократно, с различными вариациями как в аспекте ограничений решаемой задачи, так и с точки зрения представления целевой функции (ЦФ) [2-4]. Следует отметить важность ограничения, налагаемого на равномерное распределение нагрузки на ПУ[5].

Среди методов решения наиболее известен точный метод, представленный в [2]. Однако метод А.Б. Барского ориентирован на длительное время выполнения при относительно небольшой размерности исходных данных, что не всегда реализуемо.

Задача формирования конфигурации ИУС может решаться как в условиях проектирования ИУС (ограничения на время поиска решений формально отсутствуют), так и в процессе восстановления системы после сбоев, когда необходимо получить любое допустимое решение. И в первом, и во втором случае в настоящее время популярны такие методы алгоритмической оптимизации, как генетические алгоритмы (ГА) [6] и имитация отжига (ИО) [7-9]. Следует отметить, что научное сообщество до сих пор так и не пришло к решению, какой метод более эффективен. Учитывая разнообразие параметров настройки ГА и ИО, а также приоритеты, определяемые для оценки эффективности алго-

ритмов, оценка эффективности группы алгоритмов или метода является нетривиальной комплексной задачей. Для получения оценок эффективности ГА и ИО в качестве критерия эффективности выберем качество решения, получаемое за относительно небольшое количество вызовов ЦФ (для метода ИО сокращение количества вызовов ЦФ возможно за счет использования, например, законов изменения температуры по схеме «тушения»). Так как вычисление ЦФ и проверка ограничений наиболее трудоемко в случае распределения подзадач по ПУ, будем полагать, что в случае дефицита времени нам необходимо ограничиться малым количеством вызовов.

Экспериментальное сравнение эффективности ГА и ИО

Ниже представлены результаты экспериментального исследования синтезированных алгоритмов ИО и ГА для формирования конфигурации ИУС.

На рис. 1 видно, что для малого количества вызовов ЦФ алгоритм ИО с построением расписания с использованием стратегии минимизации простоев ПУ позволяет получить результаты лучше, чем ГА с аналогичной стратегией построения расписания. Та же тенденция наблюдается на рис. 2-4.

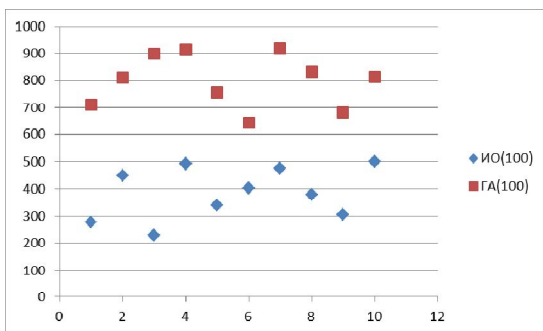


Рис.1. Решения, получаемые при 100 вызовах ЦФ для ИО и ГА

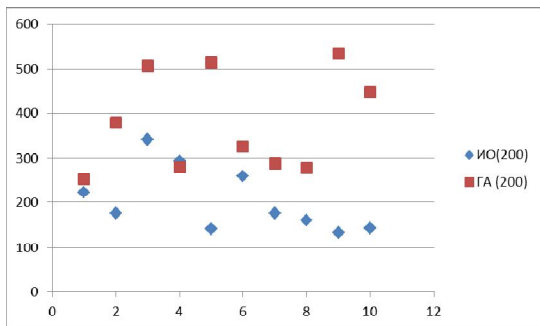


Рис.2. Решения, получаемые при 200 вызовах ЦФ для ГА и ИО

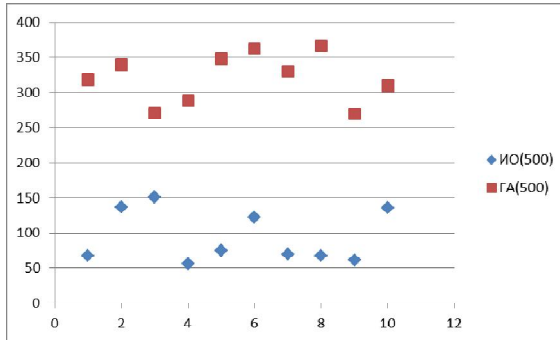


Рис. 3. Решения, получаемые при 500 вызовах ГА и ИО

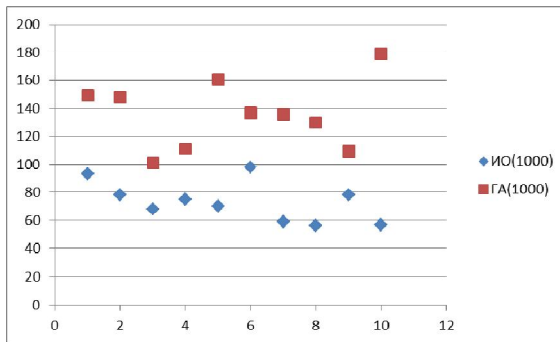


Рис. 4. Решения, получаемые при 1000 вызовах ЦФ для ГА и ИО

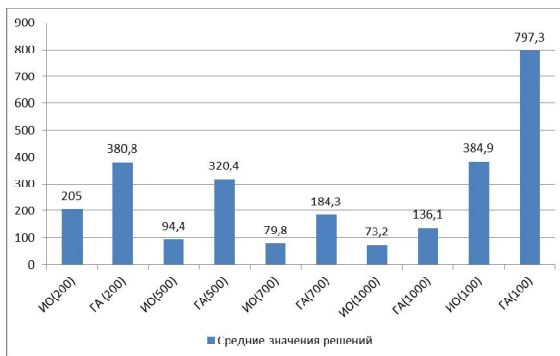


Рис.5. Средние значения получаемых решений

Видно, что с увеличением числа вызовов ЦФ растет качество решений, получаемых ГА; возможно, при дальнейшем росте количества итераций результат работы ГА будет превосходить результат работы ИО. Также на рис.1-4 видно, что при равном количестве вызовов ЦФ ИО превосходит ГА в рамках поставленного эксперимента.

На рис. 5 видно, что с увеличением количества вызовов ЦФ средние значения получаемых решений имеют тенденцию к уравниванию.

Выводы

Проведенный эксперимент позволяет сделать заключение о том, что в условиях ограничений на время вычислений метод ИО (от 100 и более вызовов ЦФ) имеет преимущества перед ГА, позволяя при равных количествах вызовов ЦФ получать лучшие решения. Следует отметить, что эксперимент проводился для алгоритмов, реализованных централизованно, однако это обстоятельство не умаляет значимости результатов: параллельные схемы ИО предполагают запуск на различных ПУ экземпляров алгоритма. Следовательно, в случае выбора лучшего решения, при равном времени повышаются шансы обнаружения «лучшего» результата.

1. Каляев И.А, Мельник Э.В. Децентрализованные системы компьютерного управления. Ростов н/Д: Издательство ЮНЦ РАН, 2011. 196с.
2. Барский А.Б. Параллельные процессы в вычислительных системах: планирование и организация. – М.: Радио и связь, 1990.
3. Костенко В.А. Возможности генетических алгоритмов для решения задач синтеза архитектур и планирования параллельных вычислений// Труды Третьей Международной научной конференции “Дискретные модели в теории управляющих систем” (Красновидово’98) М.: Диалог МГУ, 1998. - С.53-58.
4. Костенко В.А. Задача построения расписания при совместном проектировании аппаратных и программных средств// Программирование, 2002., №3.
5. Клименко А., Мельник Э. Задача реконфигурирования распределенных информационно-управляющих систем // В мире научных открытий. №6(42). 2013. С. 245-260.
6. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. - М.:ФИЗМАТЛИТ, 2003. - 432 с. - ISBN 5-9221-0337-7.
7. Лопатин А. С. Метод отжига // Стохастическая оптимизация в информатике. СПб.:Изд-во СПбГУ, 2005. Вып. 1. С. 133–149.
8. F. Buseti, Simulated annealing overview. 2003. (<http://citeseer.uark.edu:8080/citeseerx/viewdoc/summary;jsessionid=9F958F3AD8ACB341A84315A737883592?doi=10.1.1.66.5018>)
9. L.Ingber, Simulated annealing: practice versus theory.1993. (<http://citeseer.uark.edu:8080/citeseerx/viewdoc/summary?doi=10.1.1.15.1046>)

ОЦЕНКА БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
vvk@mvs.sfedu.ru, sai@mvs.sfedu.ru*

В настоящее время все большее применение как в науке, так и в технике получают распределенные вычислительные и управляющие системы. К таким системам можно отнести: системы автоматического регулирования и управления техпроцессами, телекоммуникационное оборудование, центры хранения и обработки больших массивов данных.

Внедрение таких систем позволяет значительно упростить работу персонала, существенно повысить технологичность, а значит снизить себестоимость и повысить эффективность производственного процесса. Однако внедрение новых технологий, совершенных на сегодняшний день вычислительных и управляющих систем все сильнее обостряет противоречия между новыми средствами производства и традиционными способами их использования, между современными проблемами безопасности и традиционными подходами к обеспечению проектной и производственной безопасности, что особенно актуально для техногенных промышленных объектов.

Повышение защищенности критически важных объектов сферы предъявляет высокие требования к функциональной безопасности используемых систем и особенно встроенного в них программного обеспечения, так как с развитием информационных технологий значительно повысились угрозы техногенных катастроф из-за сопутствующих вредных факторов или преднамеренного нанесения ущерба.

Еще в 1975 году в статье “The Protection of Information in Computer Systems” [1] были сформулированы принципы, которые сегодня имеют еще большую актуальность при создании программного обеспечения:

простота механизмов (economy of mechanism) - архитектура должна быть как можно более простой и компактной;

безопасность по умолчанию (fail-safe defaults) - решения о предоставлении доступа необходимо принимать на основе назначения прав, а не их исключения;

полный контроль (complete mediation) - предоставление доступа к любому объекту должно осуществляться только после проверки наличия соответствующих прав;

открытая архитектура (open design) - ее не следует держать в тайне;

разделение полномочий (separation of privilege) - механизм защиты, требующий двух и более ключей;

минимум привилегий (least privilege) - любая программа и любой пользователь системы должны иметь минимально необходимый набор привилегий;

сокращение количества общих механизмов (least common mechanism) - механизмов, доступных или зависимых более чем от одного пользователя;

психологическая приемлемость (psychological acceptability) – простота и понятность человеко-машинного интерфейса.

Ужесточение требований к безопасному функционированию встроенного программного обеспечения привело к тому, что в настоящее время появилось несколько моделей создания безопасных программных систем, которые полностью учитывают вышеперечисленные принципы. Такими широко используемыми моделями проектирования являются: *командная разработка программного обеспечения* и *разработка программного обеспечения методом «стерильной комнаты»*.

Командная разработка программного обеспечения (Team Software Process, TSP) представляет собой использование при командной разработке определенного набора операций. На этапе проектирования необходимо определиться со спецификацией требований и отразить в ней требования к безопасности. На этапе реализации необходимо обеспечить точное соответствие создаваемого программного обеспечения установленным требованиям, исключая не документированные возможности. В модель командной разработки безопасного программного обеспечения (TSP-Secure) добавляются дополнительные процедуры обеспечения безопасности. Разработчики проходят дополнительную подготовку в области защиты, изучая общие причины возникновения уязвимых мест, методы проектирования, ориентированные на безопасность, и способы их реализации и тестирования [2].

Программное обеспечение, построенное с использованием данной модели, имеет около **0,06 дефектов** проектирования и реализации на каждую тысячу строк кода.

Разработка программного обеспечения методом «стерильной комнаты» представляет собой процесс создания и сертификации корректных программных систем со статистическим контролем качества. Он охватывает все этапы разработки, включая определение спецификаций требований и архитектуры, проверку корректности функционала, а также статистическое тестирование. При разработке с помощью данной

модели между специалистами разделяются задачи: определение спецификаций, разработка, тестирование, сертификация.

Программное обеспечение, построенное с использование данной модели, имеет около **0,1 дефектов** проектирования и реализации на каждую тысячу строк кода [3].

Следующими не менее важными моментами при создании безопасного программного обеспечения являются тестирование и верификация (анализ). При тестировании и верификации создаваемого программного обеспечения применяются различные технологии, среди которых можно выделить пять основных, используемых в настоящее время поставщиками программной продукции.

Экспертиза (software review) – представляет собой набор методов верификации и тестирования, в которых анализ уязвимостей программного обеспечения выполняется людьми (экспертами). Обычно выделяют: техническую экспертизу (technical review), сквозной контроль (walkthrough), инспекция (software inspection), аудит (audit). Применение этой технологии может обнаружить от 50 до 90 % ошибок ПО, что существенно снижает затраты за счет их раннего обнаружения. Однако для ее качественного проведения необходим персонал с соответствующей квалификацией, уровнем образования, подготовкой и опытом.

Статический анализ – анализ программного обеспечения, производимый без реального выполнения исследуемых программ. В зависимости от используемого инструмента глубина анализа может варьироваться от определения поведения отдельных операторов до анализа, включающего весь имеющийся исходный код. Способы использования полученной в ходе анализа информации также различны – от выявления мест, возможно содержащих ошибки, до формальных методов, позволяющих математически доказать какие-либо свойства программы.

Динамический анализ – анализ программного обеспечения, выполняемый при помощи выполнения программ на реальном или виртуальном процессоре. В качестве методов выделяют: тестирование, имитационное тестирование (модульное, интеграционное, системное), мониторинг.

Формальные методы – используют для анализа свойств ПО формальные модели требований, поведения ПО и его окружения.

Синтетические методы – представляют собой комплексирование различных методов и техник, представленных выше. Они пытаются взять лучшее из различных методов, отбросив их недостатки, что является наиболее предпочтительным при создании встроенного программного обеспечения систем для важных объектов техносферы. Синтетические методы достаточно молоды, они получили существенное развитие в последние 10-15 лет. Среди них стоит выделить тестирование на осно-

ве моделей (model based testing) и мониторинг формальных свойств (runtime verification, passive testing). Однако уже сейчас существуют готовые средства, автоматизирующие их процесс.

Учитывая вышеизложенное, для достижения приемлемого (заданного) уровня функциональной безопасности при проектировании встроенного программного обеспечения вычислительных и управляющих систем критически важных объектов техносферы необходимо правильно выбрать и использовать как модель проектирования, так и технологию тестирования и верификации.

При выборе моделей проектирования необходимо учитывать, что каждая из моделей проектирования имеет свои сильные стороны и недостатки, поэтому при выборе основным критерием является отношение общей стоимости проекта и требований по безопасности программного обеспечения при его функционировании. Либо это будут большие финансовые и временные затраты, но наименьшее количество дефектов в программном продукте при использовании первой модели проектирования, либо меньшее время и гибкий подход к проектированию, но большее количество дефектов в программном продукте при использовании второй модели проектирования.

При выборе технологий тестирования и верификации готовых проектов следует учитывать, что в распределенных вычислительных и управляющих системах сложность при проектировании и тестировании создает их архитектура и большое количество взаимодействующих между собой программных продуктов. Поэтому для них предпочтительно применять методы динамического анализа, с помощью которых можно проверить реальную работу системы в реальном окружении, что невозможно сделать с помощью других технологий. Это дает возможность получить данные о функциональности, безопасности, производительности и надежности ПО.

1. J. Saltzer, M. Schroeder. The Protection of Information in Computer Systems. Proc. IEEE, vol. 63, no. 9, 1975.
2. N. Davis, J. Mullaney. The Team Software Process in Practice: A
3. Summary of Recent Results, tech. report CMU/SEI-2003-TR-014, Software Eng. Inst., Carnegie Mellon Univ., September 2003.
4. R. Linger, S. Powell. Developing Secure Software with Cleanroom
5. Software Engineering. 2004.
6. IEEE 829-1998. Standard for software test documentation. New York: IEEE, 1998.

РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ДЛЯ ВЫСОКОРИСКОВЫХ ПРОИЗВОДСТВ

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
kypovykh@gmail.com*

Инженерный анализ систем управления, внедрение прогрессивных технологий в производство, разработка новых автоматизированных систем и управление оборудованием на базе современной вычислительной техники являются важными задачами современного машиностроения.

В данный момент существует проблема, состоящая в том, что на большинстве объектов высокорискового производства используются устаревшие интегрированные системы управления, в которых один мощный процессор управляет большим количеством пассивных периферийных устройств. Заменой таких систем являются распределённые системы управления, когда каждый элемент системы является активным устройством. Такие системы больше всего подходят для высокорискового производства.

В интегрированных системах используется мощный процессор с большим количеством пассивных устройств, а передача информации осуществляется по параллельной шине (например VME, PCI, ISA, PC/104 и т.д.).

Такая организация системы имеет следующие недостатки:

- необходимость применения мощных процессоров;
- большие трудности, связанные с расширением системы;
- низкая надежность;
- большие сроки разработки.

Стыковка элементов распределенной системы управления осуществляется с помощью стандартных сетевых интерфейсов на аппаратном уровне и с помощью стандартных протоколов – на программном. Элементом системы может быть плата универсального контроллера, содержащая стандартный сетевой интерфейс или промышленный контроллер с набором модулей или плат.

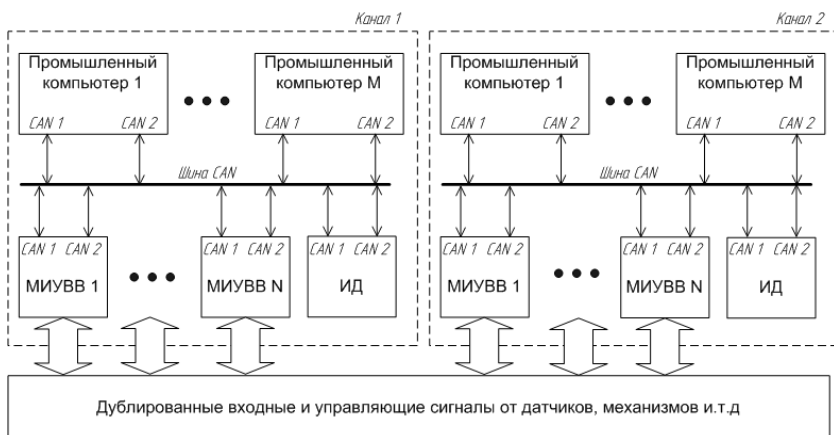
Распределенные системы обладают следующими преимуществами:

- легкая расширяемость;
- высокая надежность;
- малые сроки разработки;
- легкость тестирования и отладки;

- возможность распределения системы по объекту;
- использование компьютеров и контроллеров меньшей мощности.

В распределенных системах управления используются промышленные компьютеры разных исполнений, в зависимости от места установки, но зачастую все они имеют Ethernet- и CAN-интерфейсы. В распределённой системе управления узлы и электронные блоки могут быть скомпонованы в одном или нескольких конструктивах разного формата, что обеспечивает гибкую архитектуру системы. Например, в системе с большим количеством датчиков и исполнительных устройств распределение системы по объекту позволяет существенно уменьшить количество кабелей, повышая её надёжность и простоту эксплуатации.

Для того чтобы оборудование отвечало современным требованиям производства, необходимо использовать новейшую элементную базу, не содержащую аналоговые компоненты. Такой подход позволит исключить постоянную подстройку при длительной работе и расширить температурный режим работы. Поэтому необходимо оснащать микроконтроллером каждую составную часть оборудования с «привязкой» к цифровой промышленной цепи.



Структурная схема распределенной системы управления полярным краном

На рисунке отображена разработанная распределённая система управления полярным краном. Основными элементами данной системы являются: промышленные компьютеры со встроенным CAN-

интерфейсом, модули интеллектуального удалённого ввода-вывода (МИУВВ) и интеллектуальные датчики (ИД) в соответствии с ГОСТ Р 8.673-2009. Такая система имеет ряд преимуществ:

- два канала для управления краном, в случае если один выйдет из строя;
- входные и управляющие дискретные сигналы дублированы и обрабатываются разными МИУВВами, что делает данную распределенную систему управления реконфигурируемой;
- каждое устройство имеет два канала CAN для повышения надежности передачи данных.

За счёт того, что в системе дублируются все сигналы, а линии связи CAN также зарезервированы, данная распределённая система является ещё и легко реконфигурируемой. При выходе из строя одного или более элементов системы все его функции автоматически берёт на себя следующий подобный элемент. Это позволяет не тратить время на перекоммутацию и позволяет не останавливать производство в критические моменты работы. Каждый из контроллеров может иметь дополнительные пассивные модули ввода-вывода, которые подключаются через системную шину контроллеров. Максимальное количество модулей $N=255$, т.к. сеть CAN поддерживает до 256 устройств.

Приведенная структура распределенной системы управления позволяет повысить надежность работы полярного крана и может быть взята за основу при создании как больших, так и небольших, относительно недорогих, но достаточно мощных и гибких систем промышленного и научного назначения.

1. Распределенные системы управления [Электронный ресурс]. - <http://www.kaskod.ru/article/02syscontr/> - статья в интернете.
2. Эндрю Таненбаум, М. ван Стеен. Распределенные системы. Принципы и парадигмы // Питер. Классика Computer Science. ISBN 5-272-00053-6, 0-13-088893-1; 2003 г.
3. Государственная система обеспечения единства измерений. Датчики интеллектуальные и системы измерительные интеллектуальные. Основные термины и определения. ГОСТ Р 8.673-2009

ПОДХОД К АНАЛИЗУ СВЕРХБОЛЬШИХ СНИМКОВ СЕТЕВОГО ТРАФИКА С ПОМОЩЬЮ MAP/REDUCE

*ФГБОУ ВПО «СПбГПУ», г. Санкт-Петербург,
laboshin@neva.ru, lukash@neva.ru*

Введение

На основе анализа содержимого, а также статистики сетевого трафика можно идентифицировать различные действия в сети и получить информацию об устройстве программ, информационных систем и сетей [1]. Маршрутизаторы, а также многие иные коммуникационные устройства имеют встроенные функции для сбора разнообразной статистики. Распространенным форматом сбора статистики является Netflow. Он предусматривает запись сведений о каждом «потоке» (англ., flow), который представлен в сети в виде серии сообщений, объединенных совокупностью IP-адресов, портов и номеров протокола. Сбор такой статистики позволяет установить факт обращения определенного узла (идентифицированного IP-адресом) к другому узлу, время обращения, количество переданного и полученного трафика, протокол, номера портов с обеих сторон и не дает доступа к содержимому трафика. При этом полное архивирование сетевого трафика может оказаться чересчур объемным. Например, при пропускной способности сетевого канала в 10 Гбит/с за 12 часов будет передано в одну сторону более 50 Терабайт информации. Обработка такого объема информации не может быть произведена одним сервером за приемлемое время, поэтому полное архивирование всего трафика ведется лишь в редких случаях и не для всех протоколов. Однако в соответствии с приказом Минкомсвязи, обязующим Интернет-провайдеров предоставлять снимки сетевого трафика за последние 12 часов, важность оперативной обработки таких снимков является неоспоримой. Для анализа данных такого объема необходима вычислительная система, в которой система хранения и вычислительные ресурсы могут быть легко масштабированы в зависимости от размера входных потоков данных. Такие возможности могут быть получены посредством использования облачных вычислительных систем и распределенных кластерных файловых систем. Программная модель MapReduce, разработанная компанией Google в 2004 году, позволяет параллельно обрабатывать данные большого объема путем задания функций Map и Reduce [2]. Вычислительные элементы, реализующие эти функции называются, соответственно, маппер (mapper) и редьюсер (reducer). Одновременно над одной задачей может работать большое количество мапперов и редьюсеров, распределенных по машинам класте-

ра. Данные для обработки с помощью MapReduce должны быть представлены в формате ключ - значение $\langle k; v \rangle$.

Весь объём входных данных разбивается на фрагменты определённого размера или блоки; каждый такой блок поступает на вход одному из мапперов. Входящая пара $\langle k_{in}, v_{in} \rangle$ некоторым образом преобразуется в промежуточную пару $\langle k_{int}, v_{int} \rangle$. Затем промежуточные данные, полученные со всех мапперов, некоторым образом группируются по ключу k_{int} и поступают на вход редьюсерам в виде $\langle k_{int}, list\ v_{int}^j \rangle$. Таким образом, значения, соответствующие одному ключу, попадают в один редьюсер. После окончательном обработки на выходе редьюсера получаем пары $\langle k_{out}, v_{out} \rangle$, которые записываются в выходной файл. Схематично работу MapReduce-задачи можно представить рисунком 1. В данном докладе мы предлагаем модель облачной программной системы, основанной на парадигме MapReduce для параллельной обработки снимков сетевого трафика.

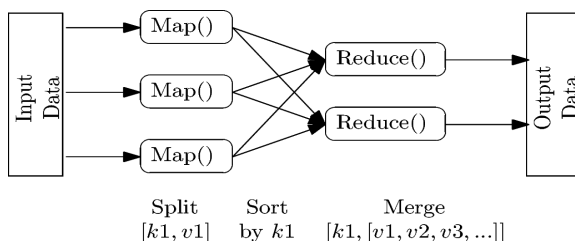


Рис. 1. Функции Map и Reduce

Метод обработки снимков сетевого трафика в распределенной вычислительной системе

Одним из самых распространенных форматов хранения снимков сетевого трафика является PCAP (англ. Packet CAPture). Библиотека libpcap является платформонезависимой библиотекой с открытым исходным кодом (версия для Windows носит название winpcap). PCAP используется в качестве основного формата такими программными средствами как tcpdump, wireshark, nmap и является де факто стандартом для захвата и анализа сетевых пакетов данных. Libpcap — бинарный файл, состоящий из глобального заголовка, позволяющего его идентифицировать и записей для каждого захваченного пакета (рис. 2)

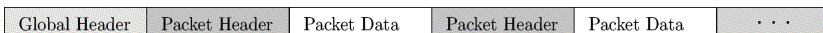


Рис. 2. Структура файла PCAP

Программы, построенные на программной модели MapReduce, в большинстве своем используются для обработки больших объемов текстовых файлов, таких как лог-файлы или веб-страницы, поэтому основной формат входных файлов – текстовый. Каждая строка интерпретируется как пара $\langle k_m, v_m \rangle$ – ключ это смещение от начала файла, значение – содержимое строки. Двоичный формат PCAP не имеет отметок между пакетами, таких как символ перевода строки в текстовых документах. Так как исходный файл разделяется на блоки фиксированной длины, пакетная запись в файле часто находится в двух соседних блоках. Длина записи также варьируется от пакета к пакету, что осложняет их выявление в пределах блока распределенной файловой системы, следовательно, для корректной работы системы необходим надежный алгоритм определения начала пакетной записи в блоке.

Размер кадра в сети Ethernet обычно находится в диапазоне 64-1518 байт (за исключением так называемых «Jumbo» фреймов). В заголовке пакетной записи PCAP файла присутствуют два 2-байтовых поля, первое содержит длину пакетной записи, второе длину пакета в сети. При стандартных настройках захвата сетевого трафика максимальная длина пакетной записи 65535 байт, длин пакета в сети никогда не превышает этого значения, следовательно, в соответствующих полях пакетной записи будут находиться равные значения. В данной работе предложен эвристический алгоритм, в котором эта особенность используется как своеобразная метка начала пакетной записи. Обработка блока входных данных начинается с поиска двух равных идущих подряд двухбайтовых полей, содержащих значение, соответствующее допустимому размеру кадра в сети Ethernet. Когда такие поля найдены, определяется предполагаемое место начала пакетной записи и производится дополнительная проверка его корректности, например путем сравнения поля Ethertype с допустимыми значениями (IEEE 802.3)[4]. После определения первой пакетной записи, поиск следующей осуществляется схожим образом с того места, где закончилась предыдущая (рис. 3).

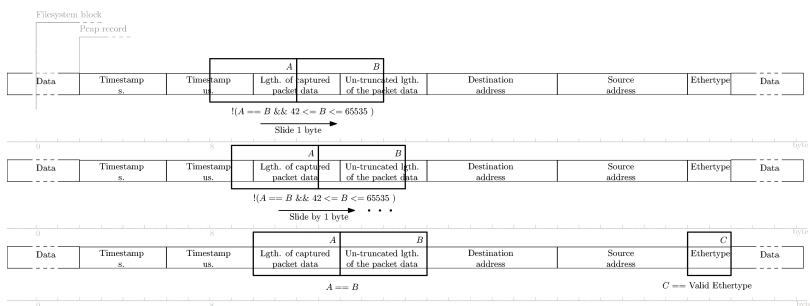


Рис. 3. Алгоритм выявления начала пакетной записи в блоке распределенной файловой системы

Сбор статистических данных сетевого трафика с использованием MapReduce

Используя описанный выше метод чтения пакетных записей в пределах блока с помощью задания простых функций подсчета, становится возможным извлечение статистической информации из файла снимка сетевого трафика, вплоть до транспортного уровня модели OSI. Результаты анализа на данном уровне предоставляют нам следующую информацию:

- общее количество трафика между отдельными подсетями (байтов, пакетов, соединений);
- общее количество локального трафика;
- выявление трафика, нарушающего политику доступа.

Задачи, которые могут быть решены на данном этапе, включают:

- выявление наличия вирусного трафика в сети (аномально большое количество входящего и исходящего трафика);
- выявление DDoS атак (аномально большое количество трафика, сгенерированного почтовыми или DNS серверами);
- доступ к запрещенным IP адресам.

Анализ содержимого сетевого трафика

Глубокий анализ пакетов сетевого трафика позволяет сопоставлять поток трафика с базой известных паттернов, известных как сигнатуры. Сигнатуры являются представлением потенциально вредоносного содержимого в виде строки символов или определенной последовательности байтов. В отличие от рассмотренных выше задач подсчета, которые могут быть выполнены на IP уровне, такой анализ не может быть проведен для отдельно взятого блока в файловой системе, а требует построения отдельных потоков данных, формируемых сетевыми приложениями в рамках информационного взаимодействия. Такие потоки, называемые виртуальными соединениями, как некоторая абстракция, существуют параллельно и друг от друга, при этом не имеют между собой разделяемых ресурсов, что позволяет осуществлять их параллельную обработку [5]. На примере протокола TCP контекст виртуального соединения можно представить в виде вектора, описывающего такие его параметры как порты и адреса источника и приемника, состояние соединения.

Для построения виртуального соединения, при обработке каждого блока с помощью mapreduce для каждого пакета задается ключ, построенный на основании некоторой коммутативной операции над хешами адресами отправителя и получателя, такой, что $F(src, dst, port) = F(dst, src, port)$. На основании значения этого ключа на стадии reduce собирается выходной файл, содержащий в себе одно независимое виртуальное соединение. Над такими файлами может быть произведен детальный анализ соответствующего протокола, например заголовков и содержимого HTTP – трафика.

Выводы

В настоящей работе предложен метод анализа снимков сетевого трафика основанный на парадигме MapReduce. Описанная система может быть использована для построения вычислительной платформы, в которой ресурсы могут масштабироваться в зависимости от объема входных данных для обработки сверхбольших файлов архивов сетевого трафика современных вычислительных сетей.

1. Федотов Н.Н. Форензика - компьютерная криминалистика // М., "Юридический мир", 2007. - 360 с. ISBN 5-91159-015-8.
2. Dean, J., Ghemawat, S. Mapreduce: Simplified data processing on large clusters. // Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation - Volume 6. pp. 10–10. OSDI'04 (2004).
3. Tcpdump & libpcap [Электронный ресурс] — Режим доступа: <http://www.tcpdump.org/>
4. IEEE 802 Numbers (2014)[Электронный ресурс] — Режим доступа: <http://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml>
5. Заборовский В.С., Лукашин А.А., Купреенко С.В., Мулюха В.А. Архитектура системы разграничения доступа к ресурсам гетерогенной вычислительной среды на основе контроля виртуальных соединений. Вестник Уфимского авиационного технического университета. Управление, вычислительная техника и информатика, 2011, No 5 (45).

А.И. Машошин, А.И. Соколов

ИНТЕГРИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ АВТОНОМНОГО НЕОБИТАЕМОГО ПОДВОДНОГО АППАРАТА*

*ОАО "Концерн "ЦНИИ "Электронприбор", г.Санкт-Петербург,
aimashoshin@mail.ru*

Одним из основных направлений развития техники для освоения океана является создание подводных роботов, получивших название автономных необитаемых подводных аппаратов (АНПА). АНПА предназначены для решения широкого круга задач как мирной, так и военной направленности [1,2,3].

Создание любых роботов представляет собой сложную научно-техническую задачу, а при создании АНПА наряду с трудностями, возникающими при создании наземных роботов, добавляются новые. К ним относятся:

* Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект 12-08-00511-а).

- необходимость функционирования АНПА в условиях высокого гидростатического давления предъявляет особые требования к прочности всех конструкций;
- отсутствие под водой таких сервисов как GPS/GLONASS существенно затрудняет навигацию АНПА;
- ограниченное расстояние, на котором возможен обмен данными с пунктом управления по гидроакустическому каналу, а также крайне ограниченная пропускная способность этого канала значительно затрудняют управление АНПА;
- ограниченный радиус и низкая разрешающая способность подводного технического зрения и связанные с этим большие трудности с распознаванием подводных объектов существенно сокращают эффективную дистанцию обнаружения объектов поиска;
- недостаточно высокая эффективность единичного АНПА вызывает необходимость выполнять миссии в составе группы АНПА, что, в свою очередь, требует реализации сложных алгоритмов управления групповым поведением АНПА и сложных технических средств, обеспечивающих реализацию этих алгоритмов.

К перечисленным техническим трудностям можно добавить высокую стоимость создания и испытаний АНПА, а также поддержания их в рабочем состоянии на протяжении всего жизненного цикла.

Целью работы является краткое описание одной из наиболее сложных составных частей АНПА – интегрированной системы управления (ИСУ).

ИСУ – это система, отвечающая за выполнение поставленной перед АНПА задачей (называемой миссией). Абсолютное большинство выполняемых АНПА миссий состоит из следующих этапов:

- переход в заданную точку;
- обнаружение в окрестности этой точки заданного подводного объекта;
- осуществление с этим объектом заданного взаимодействия;
- возвращение в исходную точку;
- доклад о выполнении задания.

Из данного перечня следует, что ИСУ для обеспечения выполнения миссии должна иметь в своем составе (рис. 1):

- канал ввода маршрутного задания, содержащего координаты объекта воздействия, его описание, цифровую карту района действия, описания текущих гидроакустических условий в районе и др.;
- канал обмена данными с системой управления (СУ) средствами движения АНПА, по которому от ИСУ передаются команды управления движением АНПА, а в обратную сторону движутся доклады об исполнении/неисполнении этих команд и о текущем состоянии средств движения (исправны/неисправны);



Рис.1. Структурная схема ИСУ АНПА

- канал обмена данными с внешними манипуляторами АНПА, по которому от ИСУ передаются команды управления их воздействием на заданный объект;
- навигационную подсистему, обеспечивающую контроль текущих координат и параметров движения АНПА в трехмерном подводном пространстве (гироскоп, акселерометры, лаг-эхолот и др.);
- подсистему радиосвязи, обеспечивающую обмен данными с надводным (воздушным) командным пунктом, а также прием сигналов GPS/GLONASS;
- подсистему гидроакустической связи, обеспечивающую обмен данными с подводным командным пунктом, а также со взаимодействующими АНПА;
- подсистему освещения обстановки по гидроакустическому каналу, обеспечивающую дальнейшее обнаружение и распозна-

вание подводных объектов, в том числе объекта поиска и навигационных препятствий;

- подсистему освещения обстановки по оптоэлектронному (телевизионному) каналу, обеспечивающую уточнение класса обнаруженного объекта и определение его положения и состояния;
- подсистему освещения обстановки по гидрофизическим каналам, обеспечивающую обнаружение подводных объектов по физическим полям, отличным от гидроакустического (магнитному, химическому, гидродинамическому, радиационному и др.);
- подсистему управления, в которой осуществляется совместная обработка информации, поступающей от всех датчиков и каналов связи и обмена данными, ее анализ, принятие управленческих решений и выдача команд в соответствующие системы АНПА на исполнение принятых решений. Данная подсистема является электронным мозгом ИСУ и всего АНПА.

ИСУ АНПА строится на базе единого цифрового вычислительного комплекса (ЦВК), к которому предъявляются достаточно жесткие требования:

- высокая производительность (сотни Гфлопс) для реализации всего перечня функций, выполняемых подсистемами ИСУ;
- малые габариты, ограниченные размерами приборного отсека АНПА (до 100 литров);
- низкое энергопотребление, величина которого ограничена емкостью малогабаритной аккумуляторной батареи, которая должна в течение длительного времени (до нескольких суток) обеспечивать энергоснабжение всех технических средств АНПА;
- высокая надёжность (резервирование каналов передачи информации).

С учётом вышеозначенных требований к ЦВК предлагается следующий его состав:

- вычислительные машины спецобработки (ВМ СО), обеспечивающие первичную обработку больших потоков входных данных в каналах освещения обстановки;
- вычислительные машины общего назначения (ВМ ОН), реализующие распределенную вычислительную сеть и обеспечивающие возможность реконфигурации и перераспределения ресурсов.
- дублированный коммутатор беспроводной сети.

Структурная схема ЦВК приведена на рис. 2. В качестве физической реализации распределенной вычислительной сети предлагается выбрать стандарт IEEE 802.11n, обеспечивающий беспроводной способ передачи

данных со скоростью до 800 Мбит/сек внутри ограниченного объема приборного отсека. Это позволит сэкономить место внутри АНПА на прокладке кабельных трасс и достаточно гибко располагать вычислительные модули в приборном отсеке АНПА, обеспечивая оптимальную компоновку ЦВК. Также наличие беспроводной сети позволит оперативно осуществлять реконфигурацию сетей и вычислительных модулей.

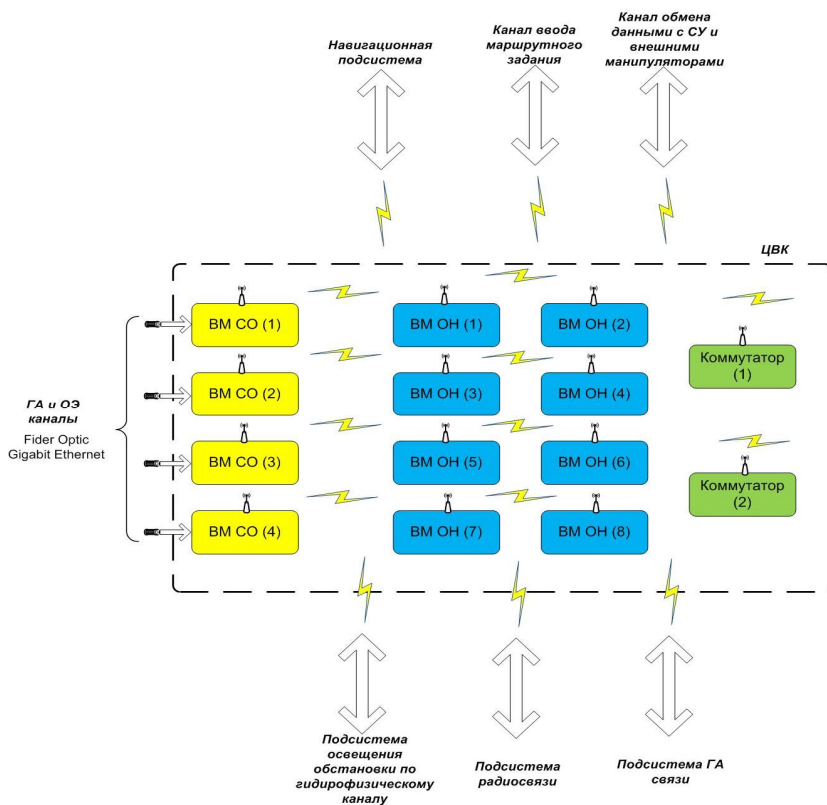


Рис.2. Структурная схема ЦВК АНПА

1. Рогозин Д.О. Заглянем в бездну. Россия приступает к освоению гидрокосмоса на новом уровне // Российская газета, Федеральный выпуск №6331 (59), 14.03.2014 г. (www.rg.ru/2014/03/14/rogozin.html).
2. Илларионов Г.Ю., Сиденко К.С., Бочаров Л.Ю. Угроза из глубины: XXI век. – Хабаровская краевая типография, 2011.
3. Береснев Валерий, Глухов Дмитрий. «Курск» 2.0» <http://flotprom.ru/publications/analytics/gvk/partone/>

НАДЕЖНОСТЬ СЕТЕЦЕНТРИЧЕСКИХ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ ПРИ РАСПРЕДЕЛЕННОМ ДИСПЕТЧИРОВАНИИ*

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог
donat.ivanov@gmail.com*

В настоящее время сетецентрический подход находит применение в различных областях деятельности, например, в атомной энергетике, в нефте- и газодобывающей отрасли [1,2], в системах подводного наблюдения [3,4] и др. [5–10]. Суть данного подхода заключается в том, что системообразующим элементом распределенной системы является коммуникационная среда, реализуемая на базе сетевых интерфейсов, и все другие устройства (устройства обработки информации, сенсоры, исполнительные устройства) взаимодействуют непосредственно через коммуникационную среду.

В работе [11] предложена концепция построения сетецентрических информационно-управляющих систем (ИУС) на основе взаимодействия локальных диспетчеров (ЛД) устройств обработки информации (УОИ), а также разработаны методы решения задачи оптимизации распределения заданий между ресурсами сетецентрической ИУС распределенных мехатронных объектов с ограничениями в виде заданных кластеров и методы организации доступа к распределенным структурированным и неструктурированным данным. Работоспособность предложенных методов организации и функционирования сетецентрических ИУС с децентрализованным диспетчированием подтверждена с помощью имитационного моделирования и экспериментов.

В данной работе проведено сравнение показателей надежности сетецентрических ИУС, построенных на базе предложенной концепции и на основе известных подходов. Рассматривались режимы парирования отказавших УОИ внутри кластера (от 1 до 3 УОИ) и режим парирования отказавшего кластера.

Эффективность новых методов и алгоритмов, выраженная в снижении аппаратных затрат на изготовление и эксплуатационных расходов, получена в ходе численных экспериментов. Сравнивались следующие варианты организации ИУС:

1. ИУС с распределенным диспетчированием (рис. 1, а).
2. ИУС с централизованным диспетчированием (1 УОИ каждого кластера выделяется под задачи диспетчирования, рис. 1, б).
3. ИУС с централизованным диспетчированием (в каждый кластер добавляется по 1 УОИ для диспетчирования кластера, рис. 1, в).

* Работа выполнена при финансовой поддержке РФФИ (проект № 14-08-00776а)

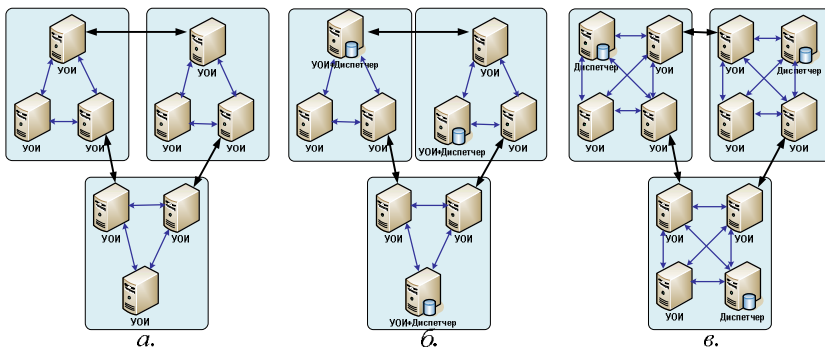


Рис. 1. Сетецентрические ИУС с распределенной диспетчеризацией и выделенным диспетчированием

ИУС состоит из кластеров, каждый из которых состоит из некоторого количества УОИ. Каждый УОИ имеет некоторую вероятность безотказной работы. В зависимости от того, сколько отказов может парировать каждый УОИ, будет зависеть вероятность безотказной работы кластера. В зависимости от этой вероятности и от того, сколько отказов кластеров может парировать ИУС, будет зависеть вероятность безотказной работы системы в целом, рассчитанной по формуле

$$P = \sum_{k=0}^B C_{H+B}^{H+B-k} P_{\text{УОИ}}^{H+B-k} (1 - P_{\text{УОИ}})^k, \quad (1)$$

где H – количество «основных» УОИ (минимально необходимое количество УОИ);

B – количество парируемых отказов ($H+B$ – общее число УОИ в системе);

$$C_{H+B}^{H+B-k} = \frac{(H+B)!}{(H+B-k)!k!} - \text{число сочетаний для } k \text{ отказов};$$

$P_{\text{УОИ}}$ – вероятность безотказной работы УОИ, вычисленная по формуле:

$$P_{\text{УОИ}} = e^{-\lambda t}, \quad (2)$$

где t – время работы;

λ – интенсивность отказов, принимаем $\lambda=10-5$ (1/ч) (типичное значение для комплектующих промышленного применения).

Применение распределенного диспетчирования при построении сетецентрических ИУС позволяет получить выигрыш не только по показателям надежности, но и по затратам на оборудование на 9-25% (в части УОИ) по сравнению с ИУС с выделенными диспетчерами при той же конфигурации. Наибольший экономический эффект достигается в ИУС с

кластерами небольшого размера. Сравнительные данные по показателям надежности (гамма-процентная наработка до уровня 0,999) и относительных затратах на вычислительные узлы (в процентах) для остальных конфигураций из пространства эксперимента приведены на рис. 2.

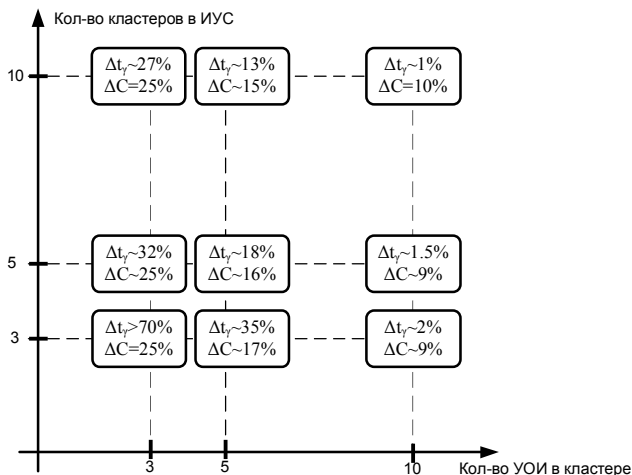


Рис. 2. Преимущества ИУС с распределенным диспетчированием с применением ЛД по сравнению с ИУС с централизованным диспетчированием по гамма-процентной наработке (Δt_γ) на отказ и стоимости вычислительных узлов (ΔC) для различных конфигураций

Расчеты вероятности безотказной работы ИУС с различным типом диспетчирования показали, что сетецентрические ИУС с распределенным диспетчированием стабильно демонстрируют лучшие показатели надежности, чем ИУС с централизованным диспетчированием. Рассматривались два варианта централизованного диспетчирования – с добавлением дополнительного УОИ в каждый кластер и без такового. ИУС с централизованным диспетчированием демонстрируют показатели надежности, близкие к ИУС с децентрализованным диспетчированием, в то время как ИУС с централизованным диспетчированием без добавления УОИ показывают значительно худшие результаты. Стоит отметить, что добавление дополнительных УОИ в каждый кластер увеличивает как стоимость ИУС, так и стоимость ее обслуживания.

Заключение. ИУС с децентрализованным диспетчированием демонстрируют более высокие показатели надежности при меньших затратах на оборудование во всем пространстве экспериментальных исследований. Наибольший эффект достигается при кластерах небольшо-

го размера (до 6 УОИ), при этом количество кластеров не имеет большого значения.

1. Коробкин В.В., Серогодский А.И. Безопасность функционирования программного обеспечения в управляющих системах на высокорисковых промышленных объектах // Шестая Всероссийская мультиконференция по проблемам управления (30 сентября – 5 октября 2013 г.) // Материалы мультиконференции в 4 т. Издательство Южного федерального университета, 2013. С. 228–232.
2. Коровин Я.С., Ткаченко М.Г., Кононов С.В. Оперативная диагностика состояния нефтепромыслового оборудования на основе технологий интеллектуальной обработки данных // Нефтяное хозяйство. 2012. № 9. С. 116–118.
3. Машошин А.И. Алгоритмы управления интегрированной сетевидной системой подводного наблюдения // Шестая Всероссийская мультиконференция по проблемам управления (30 сентября – 5 октября 2013 г.) // Материалы мультиконференции в 4 т. 2013. С. 112–116.
4. Пешехонов В.Г., Брага Ю.А., Машошин А.И. Сетевидный подход к решению проблемы освещения подводной обстановки в Арктике // Известия ЮФУ. Технические науки. 2012. № 3. С. 219–227.
5. Ефремов А.Ю., Максимов Д.Ю. Сетевидная система управления – что вкладывается в это понятие // 3-я Российская конференция с международным участием «Технические и программные средства систем управления, контроля и измерения» (УКИ-12). 2012. С. 158–161.
6. Фархадов М.П., Душкин Д.Н. Сетевидные технологии: эволюция, текущее положение и области дальнейших исследований // Автоматизация и современные технологии. 2012. № 1. С. 21–29.
7. Рахманов А.А. Принципы и подходы к концептуальному проектированию сетевидных систем // Известия ЮФУ. Технические науки. 2010. № 12. С. 125–134.
8. Чаусов Ф.. Некоторые подходы к совершенствованию системы управления войсками (силами) нового облика // Время и флот. 2011. № 4. С. 21–24.
9. Федосеев С.А., В.Ю. Столбов, Пустовойт К.С. Модель группового управления в сетевидных производственных системах // Материалы конференции «Управление в технических, эргатических, организационных и сетевых системах» (УТЭОСС-2012). Санкт-Петербург: ГНЦ РФ ОАО «Концерн «ЦНИИ «Электроприбор», 2012. С. 1240–1243.
10. Цветков В.Я. et al. Сетевидное управление промышленным предприятием // Известия высших учебных заведений. Геодезия и аэрофотосъемка. 2012. № 1. С. 106–108.
11. Иванов Д.Я., Мельник Э.В. Принципы организации децентрализованных сетевидных информационно-управляющих систем // Вестник компьютерных и информационных технологий. 2013. № 4. С. 25–30.

Э.В. Мельник, А.Ю. Таранов, М.Ю. Ольшанский, Д.В. Альбицкий

**К ВОПРОСУ ОБ ОРГАНИЗАЦИИ РЕКОНФИГУРАЦИИ
ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ
ИНТЕГРИРОВАННОЙ МОДУЛЬНОЙ АВИОНИКИ
С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА ИМИТАЦИИ ОТЖИГА**

*Государственный научно-исследовательский институт
авиационных систем, г. Москва,*

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
dvalbitzkiy@2100.gosniias.ru, MailForTexas@Mail.Ru*

В отличие от универсальных вычислительных систем, бортовые информационно-управляющие системы (БИУС) интегрированной модульной авионики (ИМА) являются основой систем управления современными летательными аппаратами. Задачи, возложенные на такие системы, выполняются с принятием решений в реальном времени, что обусловлено спецификой применения БИУС ИМА. Помимо этого, к БИУС ИМА предъявляются особые требования по надежности, связанные с непрерывностью процесса функционирования системы. При этом БИУС ИМА должна быть в состоянии самостоятельно разрешать нештатные ситуации, которые могут произойти в процессе ее работы.

Как правило, задача, решаемая распределенной БИУС, разбивается на отдельные подзадачи, распределяемые по вычислительным узлам системы и связанные между собой информационными потоками.

Перспективным путем обеспечения некоторого заданного уровня надежности БИУС ИМА является реализация устойчивости к отказам за счет реконфигурации с использованием резервных устройств (скользящее резервирование) или ресурсов. Реконфигурация подразумевает перенос в случае отказа подзадач с отказавших узлов на исправные. Однако при этом возникает ряд противоречий [1] с существующей концепцией построения систем ИМА. Например, для реконфигурации системе требуются широкие возможности в плане перераспределения подзадач с аварийного вычислительного узла на работоспособные узлы БИУС, что идет в разрез с концепцией реализации работы в режиме жесткого реального времени (все ресурсы должны быть распределены еще на подготовительном этапе конфигурации системы, т.е. до момента начала непосредственно работы БИУС ИМА). Получается, что ввиду необходимости выполнения реконфигурации требуется еще при распределении ресурсов создавать некий резерв, который будет задействован в случае отказов частей системы для восстановления работоспособности БИУС ИМА.

В рамках описанной концепции работы БИУС авторами были разработаны методы и алгоритмы, позволяющие производить построения конфигураций БИУС ИМА и формировать наборы конфигурации как для узлов системы, так и для коммуникационной среды (с учетом ее топологии). Дальнейшим этапом работ в данном направлении стала разработка улучшенных методов и алгоритмов, позволяющих получить более оптимальные варианты реконфигурации БИУС ИМА, которые позволили бы улучшить рабочие характеристики системы (например, уменьшить время реконфигурации, сократить число теряемых процессе реконфигурации «некритических» подзадач и т. п.).

В процессе работы по изучению способов улучшения эффективности методов и алгоритмов построения конфигураций БИУС ИМА были изучены две наиболее известные методики многофакторной оптимизации: генетические алгоритмы (ГА) и алгоритм имитации отжига (АИО). В процессе их анализа, а также на основании научных работ других авторов [2 – 5] был сделан вывод, о том, что АИО имеет примерно равную с ГА эффективность поиска оптимального решения задачи. При этом трудоемкость АИО меньше, чем трудоемкость ГА.

АИО использует упорядоченный случайный поиск на основе аналогии с процессом образования веществом кристаллической структуры с минимальной энергией при охлаждении [6]. Его большим преимуществом является свойство избегать «ловушек» в локальных минимумах оптимизируемой функции и продолжать поиск глобального минимума. Еще одним преимуществом является то, что даже в условиях нехватки вычислительных ресурсов для нахождения глобального минимума, АИО, как правило, выдает хоть и не самое оптимальное решение, но очень близкое к нему.

На данный момент разработано множество различных вариантов АИО, как общих, так и их специализаций для решения конкретных задач. Авторами предложен способ решения задачи нахождения оптимальных конфигураций для реконфигурирования БИУС ИМА на базе использования АИО.

Основной проблемой при использовании метода отжига является невозможность (в некоторых случаях) синтезировать функцию «энергии», которая используется как критерий оптимальности решения, полученного при использовании АИО. Собственно, основной задачей разработки варианта АИО для получения оптимальных конфигураций БИУС ИМА и является синтез такой функции.

Основными критериями для определения оптимальности конфигурации являются:

- 1) сохранение работоспособности узлов (узел не должен перегружаться подзадачами, т.е. суммарная трудоемкость подзадач, перемещенных на него не должна превышать его вычислительных возможностей);

$n = 1$, если все узлы являются не перегруженными,

$n = 0$, если все узлы являются перегруженными;

2) в процессе реконфигурации необходимо сохранить все «критические» подзадачи (их число должно остаться неизменным по сравнению с начальным (доаварийным) состоянием):

$pl = 1$, если все «критические» подзадачи удалось сохранить работоспособными,

$pl = 0$, если хотя бы одна из «критических» подзадач не может быть перерасмещена на работоспособный узел системы;

3) общее число коммуникационных каналов между узлами и их загруженность должны стремиться к минимуму:

$l = \sum L_{ij}^k \rightarrow \min$, где L_{ij}^k «ширина» канала между i -м и j -м узлами;

4) загруженность вычислительных узлов должна стремиться к среднему значению, чтобы получить равномерное распределение подзадач по узлам системы:

$z = \sum |Z_l - \sum T'/Nmax| \rightarrow \min$, где Z_l — загруженность l -го узла, T' — трудоемкость j -й подзадачи, $Nmax$ — число узлов системы.

5) в процессе реконфигурации допускается исключение из процесса работы «некритических» подзадач, однако число таких подзадач должно, по возможности, стремиться к максимуму:

$p^0 = (\sum P_m^0(t_0) - \sum P_m^0(t_1)) \rightarrow \max$, где $\sum P_m^0(t)$ — число «некритических» подзадач в системе на момент времени t ;

6) при прочих равных условиях более предпочтительным вариантом решения задачи поиска оптимального размещения подзадач будет считаться тот, который обеспечивает минимальное число временно останавливаемых (на время проведения реконфигурации) подзадач:

$r = \sum R_s^0 \rightarrow \min$, где R_s^0 — смена узла подзадачи ($R_s^0 = 1$) либо состояние, когда узел не меняется ($R_s^0 = 0$).

Таким образом, функция «энергии» представляет собой некоторый функционал, зависящий от перечисленных выше величин:

$$f = F(N, T, L, Z, P, R).$$

В качестве начального для АИО решения берется случайное размещение «отключенных» подзадач на работоспособных узлах, удовлетворяющее условиям $n \neq 0$, $p^1 \neq 0$, $p^0 \neq 0$. В процессе работы АИО генерируемые новые состояния системы, в которых не выполняются условия $n \neq 0$, $p^1 \neq 0$, $p^0 \neq 0$, отбрасываются сразу же, без оценки функции «энергии», что позволяет сократить время работы алгоритма.

Полученные промежуточные результаты позволяют сделать вывод о том, что АИО является достаточно хорошим методом получения конфигураций БИУС ИМА для проведения процесса реконфигурации. Ре-

зультаты работы АИО следует признать оптимальными или очень близкими к оптимальным.

К недостаткам АИО можно отнести лишь его относительную медлительность, однако при получении конфигураций заблаговременно (т.е. не в процессе работы БИУС ИМА) это не играет существенной роли.

В качестве следующего этапа работы по указанной тематике авторы предлагают произвести исследования в следующих направлениях:

- 1) поиск более адекватной функции, описывающей «энергию» состояния системы;
- 2) анализ «температурных» параметров АИО (значения начальной и конечной «температуры», закон изменения «температуры»);
- 3) анализ применения различных вариантов распределения случайных величин в АИО;
- 4) возможность применения различных модификаций АИО.

1. Ковернинский И.В., Мельник Э.В., Погорелов К.В., Таранов А.Ю. О проблемах организации вычислительного процесса в реконфигурируемой мультиагентной БИУС жесткого реального времени // Материалы Второй всероссийской научно-технической конференции "Суперкомпьютерные технологии" (СКТ-2012). Ростов н/Д: Изд-во ЮФУ, 2014. – С. 308-311.
2. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. - М.:ФИЗМАТЛИТ, 2003. – 432 с.
3. L. Arockiam, N. Sasikaladevi «Simulated Annealing Versus Genetic Based Service Selection Algorithms», International Journal of u- and e- Service, Science and Technology, Vol. 5, No. 1, March, 2012, p. 35-49.
4. «Computational Complexity and Statistical Physics», edited by Allon Percus, Gabriel Istrate, Cristopher Moore - Oxford University Press, New York, 2006.
5. Jukka Kohonen "A brief comparison of simulated annealing and genetic algorithm approaches", 1999, University of Helsinki, Department of Computer Science (<http://www.cs.helsinki.fi/u/kohonen/papers/gasa.html>)
6. Jason Brownlee. Clever Algorithms: Nature-Inspired Programming Recipes. <http://www.CleverAlgorithms.com>
7. Мельник Э.В., Таранов А.Ю., Ольшанский М.Ю., Альбицкий Д.В. Организация вычислительного процесса в реконфигурируемой мультиагентной бортовой информационно-управляющей системе жесткого реального времени.// Материалы 6 всероссийской мультиконференции по проблемам управления (30 сентября-5 октября 2013г.)//Материалы мультиконференции: в 4 т. – Ростов н/Д: Изд-во Южного федерального университета, 2013. Т.4. – С. 62-65.
8. Каляев И.А., Мельник Э.В. Децентрализованные системы компьютерного управления. - Ростов н/Д: Издательство ЮНЦ РАН, 2011. – 196 с.

ОПТИМИЗАЦИЯ АЛГОРИТМА ГРУППОВОГО УПРАВЛЕНИЯ БЕСПИЛОТНЫМИ ЛЕТАТЕЛЬНЫМИ АППАРАТАМИ В СОСТАВЕ ЛОКАЛЬНОЙ СЕТИ

*ОАО «Концерн радиостроения «Вега», г. Москва,
from_fn@mail.ru*

Для решения целого ряда задач в оборонной и народно-хозяйственной сферах часто применяются беспилотные летательные аппараты (БЛА). Такие задачи как фото- и видеосъемка лесных массивов, мониторинг состояния ледовой обстановки на реке и т.п. успешно реализуются одиночными БЛА. Однако обследование больших по площади объектов, таких как морская акватория и обширные лесные массивы, одиночными БЛА реализовать затруднительно. Поэтому в этих случаях применяются группы, состоящие из нескольких БЛА. При этом возникает задача управления группой БЛА, которая значительно сложнее, чем управление одиночным БЛА. Это следует из того, что помимо управления полетом и действиями собственно БЛА необходимо обеспечить определенную взаимосвязь и согласовать его действия с другими участниками группы при выполнении поставленной задачи.

Управление БЛА в составе группы может осуществляться различными способами: централизованным управлением, управлением с «лидером» и децентрализованным управлением группой БЛА, включающим стайное и коллективное управление с сетевым взаимодействием [1]. При этом одним из самых эффективных является коллективное управление с сетевым информационным взаимодействием между объектами [2].

Основным принципом сетевого взаимодействия является возможность для любого элемента получать информацию от любого другого участника группы [2]. Ряд проведенных теоретических исследований [3, 4, 5] показал возможность оптимального группового взаимодействия. Однако они продемонстрировали необходимость принятия специальных мер по предотвращению столкновений БЛА в процессе группового полета с заданной топологией.

Кроме того, следует подчеркнуть, что использование традиционных подходов к синтезу группового управления на основе классических алгоритмов [6, 7] с использованием в качестве модели состояния всей группы будет затруднено сложностью формулирования коллективного интереса и очень высокими требованиями к вычислительной производительности.

Цель работы — разработка и исследование одного из возможных вариантов группового управления БЛА при движении по требуемым

траекториям с сохранением заданной топологии со сниженными требованиями к вычислительной производительности.

В работе использовались следующие допущения:

- каждый агент (БЛА) группы самостоятельно формирует свое управление в текущей ситуации;
- формирование управления агент группы осуществляет на основе информации о цели, стоящей перед группой, о своем состоянии и состоянии других объектов;
- в качестве оптимального понимается такое управление каждым БЛА в текущей ситуации, которое вносит максимально возможный вклад в достижение общей цели, т.е. обеспечивает максимальное приращение общего функционала качества при переходе группы из текущего состояния в конечное [6, 7].

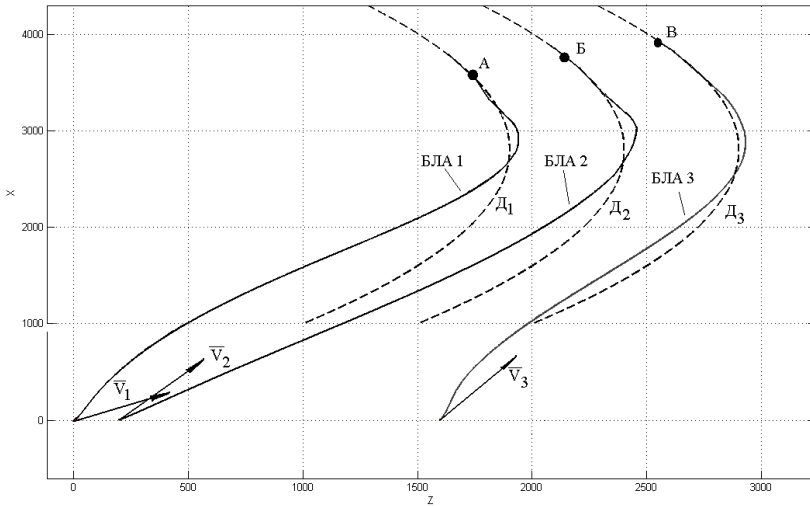
В рамках сетевого подхода на основе локальной оптимизации был получен алгоритм траекторного управления каждым БЛА группы. Принципиальным отличием полученного алгоритма от классического является возможность учета при синтезе управления i -м объектом состояния остальных j -х объектов при существенно меньших вычислительных затратах. Так, при использовании классического способа синтеза в постановке Летова-Калмана [8] для формирования сигналов управления в векторно-матричных представлениях необходимо решать систему уравнений большой размерности. Между тем, процедуру коллективного управления можно существенно упростить, если использовать оптимизацию по минимуму локального функционала качества [8], при которой каждый момент времени рассматривается как момент возможного окончания управления.

Для реализации предложенного алгоритма управления в работе на основе принятых допущений разработана математическая модель движения трех БЛА, полет каждого из которых осуществляется по законам индивидуального и коллективного управлений.

При помощи моделирования проведено исследование эффективности закона группового управления, приведено сравнение его с индивидуальным законом. В качестве показателей эффективности использовались линейные ошибки вывода БЛА в требуемые точки, ошибки управления по угловым координатам, а также время регулирования.

Пример траекторий совместного полета при случайных начальных условиях проиллюстрирован рисунком.

Результаты работы показали, что синтезированный алгоритм коллективного управления по сравнению с алгоритмом индивидуального наведения обеспечивает адекватное управление группой как по линейным, так и по угловым координатам, реализуя более высокую безопасность совместного полета, в том числе и при полете по сложным траекториям.



***Траектории полета группы беспилотных
летательных аппаратов по криволинейным траекториям***

1. Каляев И.А., Гайдук А.Р., Капустян С.Г. Модели и алгоритмы коллективного управления в группах роботов. – М.: ФИЗМАТЛИТ, 2009. – 280 с.
2. Верба В.С., Поливанов С.С. Организация информационного обмена в сетцентрических боевых операциях. // Радиотехника. 2009. №8.
3. Верба В.С., Меркулов В.И., Самодов И.О. Управление беспилотными летательными аппаратами в составе локальной сети. // Информационно-измерительные и управляющие системы. 2014. №3.
4. Харьков В.П., Меркулов В.И. Синтез алгоритма иерархического управления группой беспилотных летательных аппаратов. // Информационно-измерительные и управляющие системы. 2012. №8.
5. Гайдук А.Р., Капустян С.Г. Концепция построения систем коллективного управления беспилотными летательными аппаратами. // Информационно-измерительные и управляющие системы. 2012. №7.
6. Ройтенберг Я.Н. Автоматическое управление. – М.: Наука, 1992.
7. Брайсон А., Хо Юши. Прикладная теория оптимального управления. Пер. с англ. – М.: Мир, 1972.
8. Меркулов В.И., Дрогалин В.В., Лепин В.Н. и др. Авиационные системы радиоуправления. Т.1. Принципы построения систем радиоуправления. Основы синтеза и анализа. / Под ред. А.И. Канащенкова и В.И. Меркулова. – М.: Радиотехника, 2003.

РАСЧЕТ ПОКАЗАТЕЛЕЙ ОСУЩЕСТВИМОСТИ РЕШЕНИЯ ЗАДАЧ НА РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С НАКОПИТЕЛЕМ*

*Институт физики полупроводников им.А.В. Ржанова СО РАН,
г. Новосибирск,
pkv@isp.nsc.ru*

Введение

Исследование эффективности функционирования вычислительных систем (ВС) при решении задач различной сложности существенно зависит от ее надежности, то есть от числа работающих вычислительных узлов. Одним из вариантов повышения качества исследования является учет надежности ВС в параметре, определяющем интенсивность решения сложных задач, представленных параллельной программой.

Математическая модель

На систему массового обслуживания (СМО) поступает поток требований интенсивностью α , из которых формируется пакет объема n , $n = 1, 2, \dots$. Если пакет сформирован, то требование получает отказ. Как только СМО освобождается, она с интенсивностью $\beta(t)$ приступает к обслуживанию очередного пакета, пусть даже и не до конца сформированного. По окончании обслуживания СМО переходит к обслуживанию очередного пакета.

Требуется найти $P_k(t)$ – вероятность того, что в момент времени $t \in [0, \infty)$ пакет состоит из k нерешенных задач, при условии, что в начальный момент времени пакет был пустым; $k = 0, 1, 2, \dots$.

Имеем систему дифференциальных уравнений [1]

$$\begin{cases} \frac{d}{dt} P_0(t) = -\alpha \cdot P_0(t) + \beta(t) \cdot \sum_{k=1}^{\infty} P_k(t), \\ \frac{d}{dt} P_k(t) = -(\alpha + \beta(t)) \cdot P_k(t) + \alpha \cdot P_{k-1}(t), \quad k \in E_1^{\infty} \end{cases} \quad (1)$$

с начальными условиями

$$P_0(0) = 1, \quad P_k(0) = 0, \quad k \neq 0. \quad (2)$$

* Работа выполнена при поддержке РФФИ (гранты №№12-07-00145, 13-07-00160), Президиума РАН (проект № 18.1).

Решение системы (1) при $\beta(t) = \beta$ приведено в работе [1].

Оценка полноты накопителя

Для нахождения среднего числа нерешенных задач $M(t)$ в пакете и соответствующей дисперсии $D(t)$ используем метод моментов [2]. Система (1), с учетом (2), сведена к следующим уравнениям для математического ожидания и дисперсии

$$\begin{cases} \frac{d}{dt} M(t) + \beta(t) \cdot M(t) = \alpha, \\ \frac{d}{dt} (D(t) - M(t) + M^2(t)) + \beta(t) \cdot (D(t) - M(t) + M^2(t)) = 2\alpha \cdot M(t), \end{cases} \quad (3)$$

с начальными условиями $M(0) = 0$, $D(0) = 0$.

Пусть $\psi(N)$ – производительность системы при решении задач набора, которые забираются из накопителя, в ВС из N узлов. Тогда, если $N(t)$ – число исправных узлов в момент времени t , $t \in [0, \infty)$, то

$$\beta(t) = \frac{\psi(N(t))}{\psi(N)} \cdot \beta, \quad (4)$$

где β – интенсивность решения задач на ВС из N узлов.

Вообще говоря, $\psi(N)$ зависит от многих факторов. Например, эффективность решения задач набора зависит: от возможности параллельной обработки на системе [3]; от эффективности планировщика [4], который определяет необходимый ресурс ВС для их решения и др.

Рассмотрим один из случаев организации решения задач набора на системе в рамках построенной модели. На ВС поступает набор сложных задач, представленных параллельными программами [5, 6]. Используется планировщик для определения необходимого ресурса ВС для каждой из задач с целью минимизации общего времени решения.

На рис.1 приведены результаты эксперимента оценки эффективности (ускорения) решения задач набора на кластерных ВС в зависимости от размера системы.

Кривую, представленную на рис. 1, возможно аппроксимировать аналитической функцией следующего вида:

$$f(n) = 1 + c \cdot \exp(-a \cdot (n - 1024)^b) \quad (5)$$

при $a = 0,0034$, $b = 0.8$, $c = 1.1$.

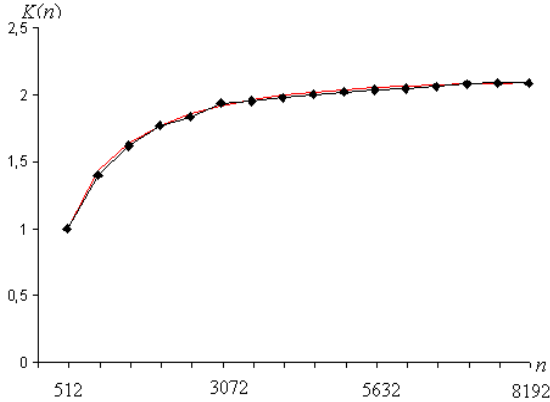


Рис. 1. Коэффициент ускорения решения задач набора на ВС.
Количество задач в наборе – 1000; максимальный ранг – 512;
 $K(n) = T_n / T_{512}$.

Алгоритм планировщика – генетический

Количество исправных машин в системе будем рассчитывать по формуле, приведенной в [1]:

$$N(t) = \frac{N\mu}{\mu + \lambda} + \frac{j \cdot \lambda - (N - j)\mu}{\mu + \lambda} e^{-(\mu + \lambda)t}, \quad (6)$$

где j – число исправных машин в начальный момент времени, λ – интенсивность отказов машин, μ – интенсивность восстановления отказавших машин [7].

На рис.2 приведены результаты расчета системы (3) с учетом (5) и (6).

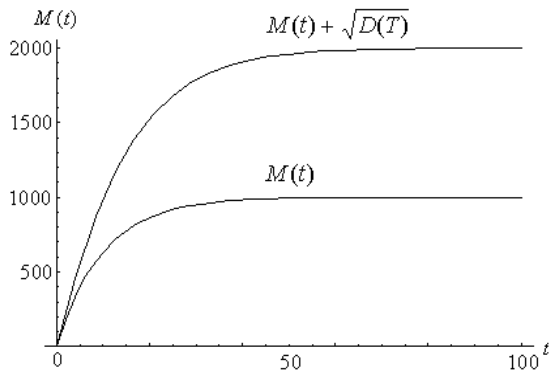


Рис. 2. Количество задач в накопителе

$$\alpha = 100 \text{ } 1/\text{ч}, \beta = 0.1 \text{ } 1/\text{ч}, \lambda = 10^{-4} \text{ } 1/\text{ч}, \mu = 0.01 \text{ } 1/\text{ч}, \beta(t) = \beta \cdot N(t) / N$$

Если считаем, что производительность системы неизменна (т.е. $\psi(N(t)) = N$), то полагаем $\beta(t) = \beta$. Тогда решение для математического ожидания и дисперсии из (3) запишется в виде

$$\begin{cases} M(t) = \frac{\alpha}{\beta} (1 - e^{-\beta \cdot t}) \\ D(t) = 2 \cdot \frac{\alpha}{\beta} \left(\frac{\alpha}{\beta} - \alpha \cdot t \cdot e^{-\beta \cdot t} \right) + 2 \left(\frac{\alpha}{\beta} \right)^2 \cdot e^{-\beta \cdot t} + M(t) - M^2(t). \end{cases} \quad (7)$$

Итак, в работе предложены аналитические решения для расчета показателей осуществимости решения задач (потока) на распределенных вычислительных системах

1. Хорошевский В.Г., Павский В.А., Павский К.В. Расчет показателей эффективности функционирования большемасштабных распределенных вычислительных систем // Вестник компьютерных и информационных технологий. Изд. "Машиностроение", №6(60), 2009. С.25-30.
2. Хорошевский В.Г., Павский В.А., Павский К.В. Расчет показателей живучести распределенных вычислительных систем. // "Вестник Томского государственного университета. Управление, вычислительная техника и информатика", №2(15), 2011г. С. 81-88.
3. Khoroshevsky V. Distributed computer systems, models and algorithms of parallel multiprogramming // Information Technologies and Systems Modelling: international conference. – France, 2010.
4. Khoroshevsky V. G., Mamailenko S. N. Stochastically optimal functioning strategies of distributed computing systems // Optoelectronics, Instrumentation and data processing, Vol. 39, No. 2, 2003, pp. 68–76.
5. Cirne, W. A model for moldable supercomputer jobs/ W. Cirne, F. Berman // 15th Intl. Parallel & Distributed Processing Symp. – 2001 (URL:<http://cseweb.ucsd.edu/~walfredo/papers/moldability-model.pdf>).
6. Dror Feitelson. Parallel Workloads Archive. (URL: <http://www.cs.huji.ac.il/labs/parallel/workload/>).
7. Хорошевский В.Г. Архитектура вычислительных систем. МГТУ им. Баумана, М., 2008. 520 с.

**ПРИНЦИПЫ ПОСТРОЕНИЯ РАСПРЕДЕЛЕННОГО
ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНОГО КОМПЛЕКСА
АВТОМАТИЗИРОВАННОГО МОНИТОРИНГА
АТМОСФЕРНО-ЭЛЕКТРИЧЕСКИХ ПАРАМЕТРОВ
ПРИЗЕМНОГО СЛОЯ АТМОСФЕРЫ**

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
n55dms@mail.ru*

Исследования процессов, протекающих в пограничном и приземном слоях атмосферы, занимают особое место в современной науке. Несмотря на многочисленные работы в этой области, остается нерешенным ряд вопросов, связанных с описанием влияния аэрозолей, радиоактивности, метеорологических факторов и солнечной активности на характеристики приземного слоя атмосферы. Измерения всех этих параметров ведутся, как правило, разрозненно и при дальнейшей обработке требуется их наложение друг на друга и отбор соответствующих данных. Для возможности одновременного сбора и обработки данных исследований необходим автоматизированный комплекс, который должен включать в себя аппаратуру и программное обеспечение для сбора, хранения и обработки атмосферно-электрических и метеорологических данных. По сравнению с существующими [1, 2], новый информационно-измерительный комплекс позволяет решить проблему совместных наблюдений на различных распределенных исследовательских станциях с автоматическим сбором информации от датчиков, их последующей передачей на сервер и обработкой. На сервере данные обрабатываются с учетом погодных условий: облачности, скорости ветра, атмосферного давления и др. После чего происходит отбор данных, удовлетворяющих требованиям "хорошей погоды", и их синхронизация. Последующая обработка и интерпретация на основе разработанных математических моделей выполняется на вычислительном кластере. Это позволяет детализировать структуру получаемых данных и разделять глобальные и локальные возмущения электрического поля.

Основными измеряемыми атмосферно-электрическими параметрами являются: напряженность электрического поля V' , удельные полярные проводимости воздуха $\lambda_{\pm}$ и плотность вертикального электрического тока j_0 . Наряду с ними проводятся специализированные измерения плотности объемного заряда атмосферы, интенсивности ионообразования, радиоактивности атмосферы, концентрации аэрозольных частиц, спектральных характеристик ионного состава, метеорологической информации и пр. Кроме упомянутых элементов, могут измеряться электрические токи осадков, заряды частиц тумана, конвективный электрический ток, ток проводи-

мости и т.д. Дополнительные измерения могут носить эпизодический характер, но именно с их помощью возможна физически правильная интерпретация получаемых данных регулярных наблюдений. Также необходимо оценить влияние солнечной активности на количественные и качественные характеристики напряженности электрического поля.

В качестве составных компонентов распределенного автоматизированного комплекса планируется использовать набор датчиков (регистраторов) и передающих устройств, а также системы сбора, хранения и обработки данных. Общая схема расположения датчиков показана на рис. 1.

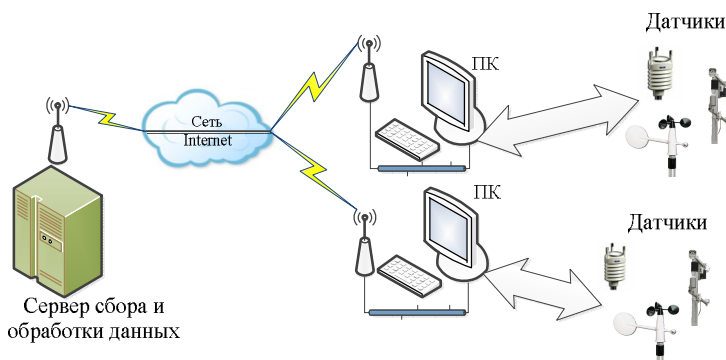


Рис. 1. Общая схема расположения датчиков

Комплекс аппаратуры, необходимый для сбора данных, включает следующие приборы: датчик напряженности электрического поля «Поле-2», датчик полярных электрических проводимостей воздуха «Электропроводность-2», датчик плотности вертикального тока, прибор для измерения интенсивности ионообразования, счетчик ядер Шольца.

Блок-схема измерительного комплекса приведена на рисунке 2. Первичные блоки приборов обычно располагаются за пределами лабораторного здания и соединяются с вторичными блоками посредством экранированных кабелей. Вторичные блоки осуществляют фильтрацию и приведение выходного сигнала к рабочим значениям. В качестве преобразователя аналогового сигнала в цифровой, можно использовать плату АЦП. Полученные с АЦП значения преобразовываются в значения измеряемых характеристик (м/с, В/м и т.д.) с использованием калибровочных коэффициентов. Профили измеряемых характеристик записываются на жесткий диск винчестера и отображаются визуально на экране монитора. После всех преобразований данные пересылаются на сервер с помощью GSM модулей. При отсутствии возможности отправить данные на сервер обработки данные накапливаются на локальном диске. В случае если приборы установлены в труднодоступных местах,

то для того, чтобы передать данные на сервер, можно воспользоваться передачей данных по сети Wi-Fi. В качестве передвижной станции можно использовать автомобиль с заранее установленной антенной Wi-Fi. После установления связи с приборами по команде оператора происходит формирование и отправка пакетов на передвижную станцию.



Рис. 2. Блок-схема измерительного комплекса

Если при перемещении передвижной станции происходит обнаружение сетей сотовой связи, то выполняется передача данных на сервер сбора и обработки данных. В случае если связь потеряна, система запоминает номер пакета с данными и прекращает отправку. При возобновлении устойчивого сигнала, отправка данных продолжается с места разрыва. На сервере происходит прием данных, синхронизация по времени, хранение и обработка. В случае, когда устройства разнесены на большие расстояния, но находятся в прямой видимости, можно использовать приборы для беспроводной передачи данных, что особенно актуально при исследованиях в горах; подобные устройства просты в настройке и эксплуатации. Однако беспроводная передача сигналов имеет определенные ограничения, такие как дальность передачи, чувствительность к преградам на пути прохождения сигнала, в редких случаях – влияние внешних радиопомех. Дальность передачи сигнала составляет от 100 метров до 3000 метров с узконаправленными антеннами на прямой видимости.

Распределенный автоматизированный комплекс мониторинга атмосферно-электрических параметров приземного слоя атмосферы позволяет установить измерительные приборы на нескольких станциях. При этом данные в системе [2] собирались с разных станций на сервер в течение нескольких суток. В новом информационно-измерительном комплексе все данные, за исключением датчиков, расположенных в труднодоступных местах и требующих присутствия передвижного комплекса, будут поступать в режиме реального времени на сервер сбора и обработки данных в течение нескольких минут. Если необходимые параметры будут измеряться сторонними организациями, то они будут использованы в процессе обработки и интерпретации данных. Таким образом, внедрение в программный комплекс современных сетевых технологий сбора и передачи данных, гибкой системы управления базами данных позволяет, в отличие от системы, рассмотренной в [3], осуществлять сбор информации в режиме реального времени, осуществлять быстрый доступ к информации с удаленно находящихся станций, хранить, обрабатывать, анализировать и интерпретировать получаемые экспериментальные и теоретические данные в любой момент времени. При анализе данных, в отличие от системы, рассмотренной в [4], учитывается влияние не только солнечной активности на состояние электрического поля, но и интенсивность ионообразования, создаваемая радоном. Полученные данные могут быть использованы: для разработки методик наземных атмосферно-электрических измерений; для улучшения методов поиска полезных ископаемых, основанных на данных об ионизации атмосферы и содержании в ней радона; для получения оценки степени аэрозольного и радиоактивного загрязнений атмосферы и разработки рекомендаций по уменьшению влияния антропогенной деятельности на электрическое состояние атмосферы.

1. Куповых Г.В., Марченко А.Г. Автоматизированный комплекс для проведения атмосферно-электрических наблюдений // Известия ТРТУ, №1, 2004. Таганрог С.203-204.
2. Алипатов М.В., Болдырев А.С., Пестов Д.А.. Интеллектуальная информационная система автоматизированного измерительного комплекса для атмосферно-электрических наблюдений // Известия ТРТУ, №6, 2009. Таганрог. С.249-253. Р-Д.
3. Марченко А. Г.. Нестационарные электрические процессы в приземном слое атмосферы : Дис.. канд. физ.-мат. наук : 25.00.30 Таганрог, 2005, 133 с. РГБ ОД, 61:05-1/928.
4. Грунская Л. В.. Оценка параметров электрического поля приземного слоя атмосферы на основе метода корреляционного приема : Дис.. д-ра техн. наук : 05.12.04, 01.04.03 Владимир, 2006, 267 с..
5. Красногорская Н. В. Электричество нижних слоев атмосферы и методы его измерения. – Л.: Гидрометеиздат, 1972. – 323 с.

ЗАДАЧА УПРАВЛЕНИЯ ПОЛЕТОМ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ В ГРУППОВОМ ПОРЯДКЕ

*ОАО «Концерн радиостроения «Вега», г. Москва,
suzansky@mail.ru*

Решение задачи управления полетом строя беспилотных летательных аппаратов (БЛА) в групповом порядке в настоящее время относится к достаточно большой научно-технической проблеме, особенно если речь идет об автоматическом управлении такими БЛА. При этом указанная задача разбивается на ряд подзадач, таких как сбор БЛА в группу, координированное управление и точное выдерживание каждым беспилотным летательным аппаратом своего места в группе, сохранение порядка строя на прямолинейных и криволинейных участках траектории полета всей группы и т.д. [1]. Решение рассматриваемой проблемы основано, как правило, на движении группы БЛА на заданной постоянной высоте. При этом стоит отметить, что при таком облете рельефа рассматриваются два ограничения: первое – необходимость выдерживания взаимного положения беспилотных летательных аппаратов в группе и, второе, невозможность использования активного радиоэлектронного оборудования беспилотных летательных аппаратов.

Цель работы – синтез одного из возможных способов коррекции траекторий движения пары беспилотных летательных аппаратов при облете препятствий, обеспечивающего безопасное движение группы.

В работе под безопасным полетом группы понимается такое движение группы, при котором обеспечивается расстояние между элементами группы не менее заданного. Для достижения поставленной цели в работе для построения закона управления группой БЛА на первом этапе формируется требуемая траектория полета каждого БЛА, на втором - формирование закона управления для каждого БЛА, обеспечивающего движение их по заданным траекториям.

Существующие способы построения опорных траекторий движения БЛА основаны на применении различных математических подходов [1]. И тем не менее, в настоящее время в основе способа облета неровностей земной поверхности группой беспилотных летательных аппаратов в горизонтальной плоскости лежит метод, который заключается в нанесении оператором контрольных точек маршрута для каждого БЛА с последующим определением соответствующих курсов полета каждого БЛА по этим контрольным точкам.

Указанная процедура отличается сложностью и большой трудоемкостью. На качество нанесения контрольных точек влияет субъектив-

ный фактор, зависящий от состояния оператора. Другими словами, два разных оператора нанесут контрольные точки по-разному, в результате чего могут быть сформированы разные траектории полета беспилотных летательных аппаратов.

Указанный метод становится особенно трудоемким и требующим высокого мастерства оператора, если приходится прокладывать маршруты двух и более БЛА для организации из группового полета. В этой ситуации оператору необходимо нанести большее количество контрольных точек, пропорциональное количеству ЛА.

В работе предлагается использовать универсальное (однотипное) описание препятствия в горизонтальной плоскости в виде описанной окружности, а требуемые траектории формировать автоматически в зависимости от начального положения ЛА (группы ЛА) на основе системы дифференциальных уравнений, которая описывает линии тока плоского потенциального потока идеальной жидкости при обтекании окружности [2].

Используемый подход построения опорных траекторий позволит решить вторую задачу, а именно синтез закона управления движением каждого БЛА в группе с требованием выдерживания заданного строя при облете препятствий в горизонтальной плоскости. Для этого в работе используется подход синтеза закона управления БЛА на основе концепции обратных задач динамики [3], при которой предполагается, что движение системы проходит по предписанной траектории. Такие траектории могут быть заданы либо аналитически, либо графически. Применительно к рассматриваемой системе из двух БЛА траекторию каждого предлагается задавать в виде системы дифференциальных уравнений, которая описывает линии тока плоского потенциального потока идеальной жидкости при обтекании эллипса, при помощи которого аппроксимируется препятствие на маршруте полета пары БЛА на постоянной высоте.

Уравнения для траекторий движения БЛА при облете препятствий синтезируются при помощи применения метода конформных отображений к построению потоков идеальной жидкости для двумерного случая. Указанные уравнения позволяют для рассматриваемого потока в любой точке пространства получить составляющие вектора скорости, которые, в свою очередь, используются для вычисления требуемого курсового угла каждого БЛА [4].

Для реализации предложенного закона управления в работе на основе принятых допущений разработана математическая модель движения пары БЛА, полет каждого из которых осуществляется по заданной траектории путем изменения курса каждого БЛА в группе на требуе-

мый. При этом для выдерживания заданного строя синтезирован закон управления скоростью каждого БЛА.

В работе рассматривалось три вида строя при полете пары БЛА: колонна, пеленг и фронт. Проведенное математическое моделирование показало удовлетворительные результаты применения синтезированного закона управления. Для исследования эффективности применения закона управления в работе исследовано влияние исходного расстояния между БЛА (до облета препятствия) на безопасность полета при облете препятствия. Показано, как изменяется указанное расстояние в процессе облета препятствия. Результаты работы позволили сформулировать требования для выбора безопасного расстояния между БЛА перед облетом препятствия.

1. Верба В.С. Авиационные комплексы радиолокационного дозора и наведения. Состояние и тенденции развития. — М.: Радиотехника. 2008.
2. Лаврентьев М.А., Шабат Б.В. Методы теории функций комплексного переменного. М.: Наука, 1965.
3. Галиуллин А.С. Методы решения обратных задач динамики. — М.: Наука, 1986.
4. Сузанский Д.Н. Метод коррекции траектории летательного аппарата при облете им зон с неблагоприятными метеорологическими условиями. // Информационно-измерительные и управляющие системы. 2011. №12.

Д.Е. Трегубов

РАСПРЕДЕЛЕННАЯ СИСТЕМА КОНТРОЛЯ И УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ПРОЦЕССОМ ПЕРЕРАБОТКИ РАДИОАКТИВНЫХ ОТХОДОВ

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
dmitry.tregubow@yandex.ru*

Интерес к распределенным системам контроля и управления возник в процессе увеличения количества и типов датчиков, площадей производства, модернизации и усложнения стандартных конструкций систем и алгоритмов управления технологическими процессами. В таких системах контроллеры, модули ввода и вывода данных, датчики, исполнительные механизмы разнесены в пространстве. Каждый контроллер работает со своими датчиками и исполнительными механизмами, с конкретной частью объекта управления, и не зависит от других контроллеров, однако взаимодействует с другими контурами и устройствами для выполнения общей задачи, достижения заданных показателей технологического процесса. Участки распределенных систем кон-

троля и управления могут находиться друг от друга на любом расстоянии, а связь между ними будет поддерживаться через несколько интерфейсов связи.

Под технологическим процессом в данном случае понимается упорядоченная последовательность взаимосвязанных действий, выполняющихся с момента начала работы системы и возникновения исходных данных до получения требуемого результата по окончании переработки радиоактивных отходов (РАО) или штатного выключения системы.

Специфика системы контроля и управления технологическим процессом переработки радиоактивных отходов предполагает разграничение групп устройств управления, сбора информации, ее первичной обработки и визуализации не только из-за значительного удаления объектов друг от друга, но и по причине функционального различия. Мониторинг состояния системы и выработка различных блокировок могут проводиться как автоматически, в режиме реального времени, так и человеком-оператором без остановки технологического процесса. Также на протекание технологического процесса минимальное влияние должны оказывать человеческий фактор, выход из строя и максимально быстрая замена устройств и составных частей системы.

Рассмотрим обобщенную структуру действующих систем управления технологического процесса переработки РАО, показанную на рисунке 1.

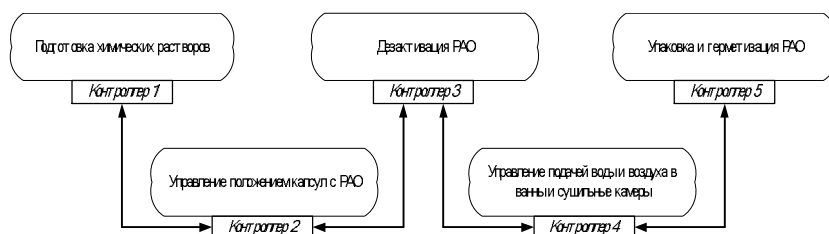


Рис. 1. Обобщенная структура действующих систем управления технологическим процессом переработки РАО

Действующие системы управления технологическим процессом переработки РАО состоят из двух уровней. Первый уровень состоит из блоков датчиков, измеряющих различные параметры объекта технологического процесса, и исполнительных механизмов, на которые подаются управляющие воздействия. На втором уровне происходит сбор с последующей обработкой информации датчиков, а также выдача различных управляющих воздействий на исполнительные механизмы. Связь между блоками технологических операций производится через контроллеры, каждый из которых поддерживает связь только с сосед-

ними контроллерами. Управление технологическими операциями происходит в полуавтоматическом режиме, а внесение изменений в процесс работы – в ручном. Для решения проблем со значительными временными задержками между получением информации, обработкой данных, формированием выходных сигналов и выдачей управляющих воздействий, а также с возможностью появления ошибок человеческого фактора, требуется разработка новых систем контроля и алгоритмов управления. Результатом решения этой задачи является разработанная распределенная система контроля и управления технологическим процессом переработки РАО. Обобщенная структура новой распределенной системы контроля и управления технологическим процессом переработки РАО показана на рис. 2.

Основным устройством сбора информации датчиков в новой системе являются локальные контроллеры (ЛК). По типу получаемой информации датчиков, выработки управляющих воздействий, а также расположения в пространстве функционально объединены несколько ЛК, либо других модулей ввода-вывода и устройств связи с объектом. Эти устройства располагаются в одном шкафу и выполняют часть технологических операций, присущих только выбранной группе.

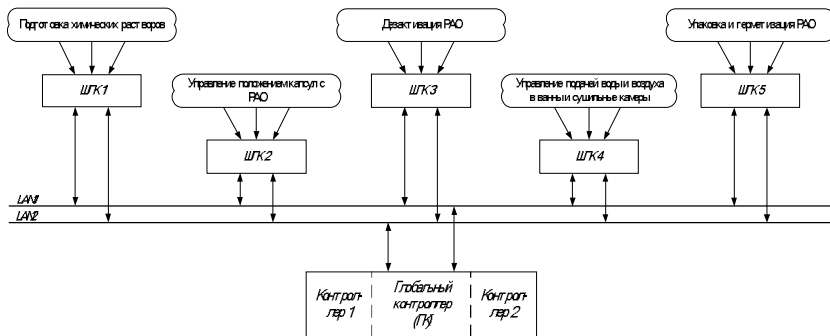


Рис. 2. Обобщенная структура новой распределенной системы контроля и управления технологическим процессом переработки РАО

Шкафы локальных контроллеров (ШЛК) различаются функционально, типом сигналов ввода вывода и разграничены в пространстве. Центром обработки, визуализации данных и выработкой управляющих воздействий выступает глобальный контроллер (ГК), который может состоять из нескольких контроллеров. ГК входит в состав программного-технического комплекса автоматизированного рабочего места оператора, расположенного на значительном удалении от зоны переработки

РАО и шкафов локальных контроллеров. Связь между ШЛК и ГК происходит по шинам данных с использованием гибридной топологии сети. Стоит отметить, что в целях безопасности и увеличения быстродействия применяется не только резервирование основных линий данных, но и разграничение шин получения данных входных сигналов и выдачи управляющих сигналов на различные периферийные механизмы.

На каждый ШЛК приходят сигналы нескольких групп датчиков. Группа датчиков представляет собой объединение датчиков с одинаковыми функциями (измерение температуры, давления, уровня жидкости и т.д.), одним уровнем сигнала (напряжением логических «0» и «1», либо диапазоном измерения аналогового сигнала) и расположенных в непосредственной близости друг от друга. Дискретные сигналы одной группы датчиков имеют напряжение более высокого уровня логической единицы, чем позволяет принимать ЛК, и требуют дополнительной интеграции в цепь устройства связи с объектом (УСО). Другие группы датчиков – аналоговые датчики тока и напряжения. Также в пределах одного ШЛК имеются группы выходных сигналов, выработанных либо ЛК, либо глобальным контроллером (ГК) для управления различными исполнительными механизмами (насосами, вентиляторами, трубопроводной арматурой и т.д.). Для связи контроллеров с исполнительными механизмами также требуется применение дополнительных УСО для повышения напряжения. Поэтому, в связи со значительным различием параметров сигналов, для сбора информации групп датчиков и выдачи управляющих сигналов на группы исполнительных механизмов распределенной системы контроля и управления технологическим процессом переработки РАО требуются несколько видов устройств ввода-вывода.

Новые методы организации сбора информации датчиков и алгоритмы распределения вычислений позволяют увеличить быстродействие и повысить отказоустойчивость. Заметное влияние оказывает распределение функций обработки и управления между двумя контроллерами в блоке ГК системы переработки РАО. В свою очередь, контроллеры в ШЛК также перераспределяют различные задачи, увеличивая скорость обработки данных и уменьшая простой системы либо при возникновении свободного времени между циклами работы, либо при выходе одного или нескольких устройств из строя.

Таким образом, применение распределенной системы контроля и управления технологическим процессом переработки РАО позволяет:

1. Повысить отказоустойчивость. В большинстве применений системы важно не допустить появления ошибок, связанных с человеческим фактором, и остановки технологического процесса.
2. Масштабировать систему. Масштабирование имеет большую важность для различных систем контроля и управления, по-

сколькx наращивание технологической базы, добавление или замена контуров и частей объекта управления происходят просто и без остановки функционирования уже подключенных решений.

3. Упростить разработку и конфигурирование системы контроля и управления. В состав системы входят множество необходимых элементов: системы программирования, библиотеки стандартных элементов, настроенные сетевые интерфейсы и протоколы.

В связи со спецификой работы, распределенная система контроля и управления технологическим процессом переработки РАО обеспечивает:

- анализ состояния системы, выработку защит и блокировок для предупреждения и ликвидации нарушения корректной работы любого механизма, а также перераспределение различных задач между контроллерами в режиме реального времени (РВ);
- визуализацию состояния системы и ведение протоколов работы для своевременного обеспечения оператора полной информацией о параметрах системы;
- прием команд от оператора с последующим изменением алгоритма работы системы без прерывания технологического процесса.
- обмен информацией глобального контроллера (ГК) с локальными контроллерами (ЛК) с частичным разграничением функций обработки и управления;

В свою очередь, локальные контроллеры выполняют следующие функции:

- проверку корректности и блокировку неверных команд, принятых от ГК во избежание ошибок управления;
- непосредственную выдачу управляющих сигналов исполнительным механизмам;
- чтение состояния и предварительную обработку информации датчиков;
- частичную выработку защит и блокировок, а также сигналов управления механизмами;
- перераспределение задач при простое;
- обмен информацией с ГК.

К основным особенностям и преимуществам новой распределенной системы контроля и управления технологическим процессом переработки РАО относятся:

- взаимозаменяемость сменных одноименных составных частей и элементов без подстройки, регулировки и остановки системы, а также наличие резервных линий съема информа-

ции особо критичных датчиков (повышения радиоактивного излучения, увеличения температуры и т.д.);

- удаление датчиков от логических контролеров (на расстояние затухания дискретного/аналогового сигнала), удаление ЛК и модулей аналогового ввода-вывода от ГК (на расстояние затухания сигнала интерфейса связи);
- сигнализация об аварийном отключении и отклонениях электрооборудования от технологических параметров световым и звуковым сигналами, визуализация нарушения на экран оператора;
- архивирование, обеспечение накопления и визуализации информации для предоставления оперативному персоналу информации о протекании технологического процесса во времени, о работе оборудования и автоматики, о действиях оператора;
- анализ состояния системы, обработка команд оператора с рабочей станции и реализация управления отдельными узлами или системой в целом, а также обеспечение однократного логического управления (срабатывания блокирований и защит) в режиме РВ;
- сбор, обработка информации датчиков различных видов, назначений и интерфейсов с выработкой управляющих воздействий на различные исполнительные механизмы (насосы, вентиляторы, трубопроводная арматура и т.д.);
- изменение технологического процесса оператором, а также перераспределение задач контроллеров между собой на любом этапе работы системы.

Таким образом, использование новой распределенной системы контроля и управления технологическим процессом переработки РАО позволило в 1,4 раза повысить надежность и на 15% увеличить эффективность при эксплуатации.

1. Хранилище твердых радиоактивных отходов III (ХТРО III) [Электронный ресурс]. - <http://sdelanounas.ru/blogs/11479/> - статья в интернете.
2. Эндрю Таненбаум, М. ван Стеен. Распределенные системы. Принципы и парадигмы // Питер. Классика Computer Science. ISBN 5-272-00053-6, 0-13-088893-1; 2003 г.
3. Пьявченко О.Н. Распределенные системы сбора и обработки информации датчиков динамических объектов // Известия ЮФУ. Технические науки. – 2011. – № 5 (118). – С. 9-15.
4. Пьявченко О.Н., Трегубов Д.Е. Микроконтроллерный коммуникационный модуль в распределенной системе сбора и обработки информации датчиков на основе гибридных топологий // Известия ЮФУ. Технические науки. – 2013. – № 5 (142). – С. 19-23.

ОПТИМИЗАЦИЯ ФОРМИРОВАНИЯ СТРОЯ ГРУППОЙ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ – КВАДРОКОПТЕРОВ*

*Уральский федеральный университет,
Институт математики и механики УрО РАН,
г. Екатеринбург
sancheZZZ@uralweb.ru sesekin@list.ru*

Постановка задачи

При решении ряда целевых задач группового применения беспилотных летательных аппаратов (БЛА) (в частности, квадрокоптеров) требуется, чтобы группа БЛА формировала и поддерживала в процессе движения строй. Заметим, что задача формирования строя группой квадрокоптеров является частным случаем задачи группового управления роботов [1]. В отношении актуальности задачи в дополнение к тому, что отмечалось в работе [2], следует отметить задачу о построении карты радиационной обстановки местности, для которой используются группы БЛА.

Пусть существует однородная группа $\mathbf{R}$ из N одинаковых квадрокоптеров $r_i \in \mathbf{R} (i = \overline{1, N})$. Состояние каждого квадрокоптера описывается некоторым вектором состояния, включающим геометрические координаты, скорости. Различные углы и т.д. Существует несколько способов задания строя, которые приведены в работе [2]. В данной работе целевое строевое положение группы будет задаваться матрицей

$$\mathbf{D}_v = \begin{bmatrix} 0 & d_{v_1 v_2} & d_{v_1 v_3} & \cdots & d_{v_1 v_M} \\ - & 0 & d_{v_2 v_3} & \cdots & d_{v_2 v_M} \\ - & - & 0 & \ddots & \vdots \\ - & - & - & 0 & d_{v_{M-1} v_M} \\ - & - & - & - & 0 \end{bmatrix},$$

где каждый элемент $d_{v_\mu v_\kappa} (\mu, \kappa \in \overline{1, M}, \mu \neq \kappa)$ матрицы $\mathbf{D}_v$ представляет собой расстояние между целевыми положениями v_μ и v_κ в искомом строю. В связи с тем, что строй задается только расстояниями между положениями v_μ и v_κ в строю, положение строя и его ориентация в пространстве могут быть произвольными.

* Работа выполнена при частичной финансовой поддержке гранта РФФИ № 13-08-00643а.

Далее рассматривается задача об определении координат целевых положений строя и распределении целевых положений между квадрокоптерами с целью минимизации времени формирования строя.

Следует отметить, что решение данной задачи сводится к определению 6 параметров: координат центра строя и углов Эйлера в оптимальном положении.

Оптимизация целевых положений между квадрокоптерами при фиксированном положении строя

Предположим, что положение строя и его ориентация в пространстве зафиксированы. Рассмотрим задачу о наискорейшем переводе группы квадрокоптеров в этот фиксированный строй из заданного начального положения рассматриваемой группы. При этом мы будем предполагать, что перемещение квадрокоптеров будет происходить по прямым с наибольшей возможной скоростью. Для решения этой задачи (задачи о назначениях) можно воспользоваться известным венгерским методом [3]. Построено соответствующее программное обеспечение.

Метод случайного поиска с линейной тактикой

Для определения оптимальных параметров строя использовались три метода оптимизации. Далее описывается метод случайного поиска с линейной тактикой [4].

1. Задаём точку. Считаем критерий в данной точке.
2. Выбираем случайное направление.
3. Находим оценку по выбранному критерию в точке, стоящей от начальной на заданном расстоянии, в выбранном направлении.
4. Сравниваем оценки, полученные в пункте 1 и в пункте 3. Если критерий в точке, описанной в пункте 3, меньше, чем в пункте 1, то выполняем пункты 3 и 4 (в качестве точки из пункта 1 выступает точка из пункта 3 на текущей итерации). Если критерий в заданной точке больше, тогда повторяем пункты 2-4.

Необходимо отметить, что критерий, о котором говорится в пункте 4, это максимальное расстояние между парами, определенными при помощи Венгерского метода, при оптимальной ориентации строя в пространстве (она также вычисляется в пункте 4).

Выполняется этот алгоритм до тех пор, пока попытки найти новое положение в точке не превысят заданного количества попыток.

При данном способе определения строевого положения решением задачи будет считаться положение, при котором центр строя будет совпадать с точкой, в которой алгоритм остановился.

Метод сведения к совокупности вложенных задач глобальной одномерной минимизации

При исследовании задачи было определено, что оптимальное положение строя находится внутри прямоугольного параллелепипеда, ограниченного минимальными и максимальными значениями по каждой из координат среди координат точек начального положения.

Алгоритм можно описать следующим способом

1. Разбиваем параллелепипед, внутри которого находится решение, на n равных параллелепипедов.
2. Для каждого из n этих параллелепипедов находим оценку по заданному критерию при помощи одного из вариантов выбора точки (точек) для определения результата (описаны далее).
3. Сравниваем получившиеся оценки и находим минимальную.
4. Выделяем параллелепипед с минимальным значением критерия.
5. Повторяем действия 1-4 для получившегося параллелепипеда.

Данный алгоритм можно повторять до тех пор, пока размеры параллелепипедов, на которые разбивается параллелепипед, не уменьшатся до необходимого размера. Таким образом, можно регулировать точность вычисления данного алгоритма. В качестве параметра, который регулирует точность, может выступить как объём, так и периметр параллелепипеда. В случае, когда необходимо ограничить число итераций, точность результата будет зависеть от числа итераций, заданных изначально.

Также следует отметить, что при любом способе реализации быстроедействие и точность получаемых результатов будет зависеть от числа n – количества параллелепипедов, на которые будет разбит начальный (и каждый выбранный далее, для следующих итераций алгоритма).

В качестве критерия (пункт 3) в нашем случае выступает максимальное расстояние между парами, найденными при помощи Венгерского метода решения задачи о назначениях.

Далее рассмотрим различные способы задания точек (точки), в которых будет рассчитываться критерий.

Одним из вариантов выбора точек является вариант, при котором в качестве точки, для которой мы будем просчитывать значение критерия, для каждого из параллелепипедов взята центральная точка фигуры.

Минусом данного алгоритма является присутствие ошибки, которая может возникнуть при условии, что существует несколько локальных минимумов. В некоторых случаях мы можем попасть в локальный минимум, который не будет являться глобальным. Следует отметить, что можно подобрать такое число n , при котором будет возможно избежать ошибки, но при этом быстроедействие программы будет снижено, и в каждом случае это число нужно подбирать “индивидуально”.

Также точки, в которых вычисляется критерий можно находить при помощи метода случайного поиска с линейной тактикой. Это позволяет несколько уменьшить вероятность ошибочно выбрать один из локальных минимумов в качестве глобального.

Данный алгоритм работает существенно дольше предыдущих алгоритмов, но при этом позволяет получить лучшие результаты.

Использование суперкомпьютера

Чтобы оценить результаты работы методов, использовался алгоритм полного перебора с заданной точностью (перебор всех положений в ограниченной области с заданным шагом, изменение углов Эйлера на заданные величины). Данный метод позволяет получить результаты с необходимой точностью. Но использование данного метода предполагает большие временные затраты при использовании обычного компьютера. Поэтому в данной ситуации было целесообразно применение суперкомпьютера.

Метод сведения к совокупности вложенных задач с выбором точек методом случайного поиска с линейной тактикой трудоемок для обычного компьютера. Его выполнение может занимать достаточно большое время при поиске оптимального положения на больших расстояниях. Поэтому использование суперкомпьютера существенно ускорило бы процесс нахождения оптимального положения. При разработке программы вычисления минимумов внутри каждого из параллелепипедов можно разделить и вычислять одновременно.

Заключение

В данной работе были рассмотрены три метода. Каждый метод имеет как отрицательные, так и положительные свойства.

С точки зрения быстродействия наилучшим является метод совокупности вложенных задач с вычислением критерия в центральной точке.

Наилучший метод, с точки зрения точности результатов, – это метод совокупности вложенных задач с выбором точек методом случайного поиска.

Случайный поиск с линейной тактикой занимает промежуточное положение, по рассматриваемым параметрам. Он показывает хорошие результаты в быстродействии и точности вычислений.

1. Каляев И.А., Гайдук А.Р., Капустян С.Г. Модели и алгоритмы коллективного управления в группах роботов. – М.: Физматлит, 2009. 278 с.
2. Иванов Д.Я. Формирование строя группой беспилотных летательных аппаратов при решении задач мониторинга // Известия ЮФУ. Технические науки. 2012. Т.4. С.219-224.
3. Зайченко Ю.П. Исследование операций. Киев: «ВИЩА ШКОЛА». 1975. 320 с.
4. Растрингин Л.А. Адаптация сложных систем. Рига «ЗИНАТНЕ». 1981. 394 с.

РАЗДЕЛ 5

Применение суперкомпьютерных технологий в науке, технике и промышленности

М.О. Бахтерев, П.А. Васёв

ОДИН МЕТОД ВИЗУАЛИЗАЦИИ РЕЗУЛЬТАТОВ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛЕНИЙ

*Институт математики и механики УрО РАН, г. Екатеринбург,
mob@0xfb.imm.uran.ru*

Введение

Результаты многих вычислительных экспериментов при визуализации несут образность, нетипичную для стандартных средств визуализации. Например, в таких средствах нет механизма визуализации «левого желудочка сердца» или «кристаллической решетки TiO_2 ».

В этих случаях приходится разрабатывать свои средства визуализации или адаптировать имеющиеся. Оба варианта сопряжены со значительными затратами.

Авторы в течение ряда лет исследуют задачу минимизации подобных затрат [1-3].

В настоящей работе предлагается оригинальный метод описания сцен трехмерной визуализации. Главная цель и особенность метода – высокая скорость, с которой можно получать визуализацию результатов различных вычислений.

Метод позволяет описывать устройство визуальных сцен на специальном языке и получать соответствующую визуализацию. На рис. 1 представлен пример работы метода.

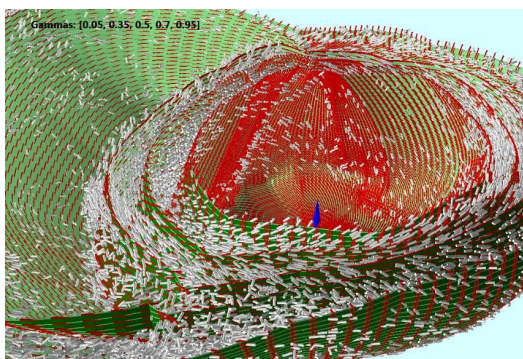


Рис. 1. Визуализация результатов численного моделирования левого желудочка сердца и экспериментальных данных [4]

Общая схема

Пользователь описывает трехмерную сцену для визуализации как набор объектов. Объекты задаются своими данными (например координатами) и свойствами отображения, такими как цвет и прозрачность.

Описание формируется на специальном языке, среди свойств которого: возможность задавать строковые значения и массивы, загружать внешние файлы и выполнять подпрограммы.

Это описание поступает на вход специальной программы, которая интерпретирует его и предоставляет различные интерфейсы для запросов к сцене.

Далее запускается программа визуализации. Она посредством запросов выясняет содержание сцены и отображает его.

Такой подход позволяет варьировать язык описания сцены, программу-интерпретатор, протоколы взаимодействия и программы визуализации. Конкретизируем эти компоненты.

Язык описания сцены

Язык текстовый, предназначен как для набора пользователем, так и для генерации вычислительными программами.

Каждый элемент данных – именованный. Имена – произвольные строковые. В имени могут быть точки.

Конкретный смысл имен, правила именования и содержания зависят от целевого интерпретатора. Например, в системе визуализации SharpEye объект сцены должен начинаться со слова «scene.». У объекта сцены могут быть подобъекты: *trimesh* – массив индексов треугольников, и *trimesh.nodes* – координаты вершин. Также допускается: *lines* – массив отрезков и *spheres* – массив сфер.

На следующих рисунках представлены элементы языка.

```
set a = 1
set a.b = 3

get a # напечатает «1»
get a.b # напечатает «3»
```

Рис. 2. Указание простых значений

```
set c = array 3 3
0 1 2
1 0 4
2 1 3
```

Рис. 3. Массивы

```

set a = 5
set b = &a
get b # напечатает «5»
set a.c = 3
get b.c # напечатает «3», т.к «b» это ссылка на «a» и на его «подобъекты»

```

Рис. 4. Ссылки

```

# включение файла данных
set d1 = array 5 3 <<file somedata.txt

# запуск программы
set d2 = array 5 3 <<exec mycalc.exe -max=5 input.dat

```

Рис. 5. Загрузка файлов и запуск программ

Рис. 5 нуждается в дополнительном комментарии. Когда в описании встречается ссылка на файл или внешнюю программу, то не происходит немедленной загрузки или вызова этой программы. Вместо этого факт такой ссылки запоминается, и фактическая загрузка будет произведена только по обращению к данному имени объекта. Так, для рис. 5 по команде «get d1» будет проведена загрузка файла somedata.txt и формирование на его основе массива размером 5 строк на 3 колонки.

Примеры описания сцен

На рис. 6 в сцене задано два визуальных объекта – массив сфер и массив отрезков. Кроме координат, задающих форму объектов, указаны и дополнительные параметры – радиусы и цвет.

```

set scene.mygraph.spheres.radius = 0.3
set scene.mygraph.spheres = array 5 3
0.3 0 0
1 0 2
-3.2 2 1
5 4 7
3 2.5 5
set scene.otherside.lines.radius = 0.1
set scene.otherside.lines.color = purple
set scene.otherside.lines = array 4 3
0.3 0 0
1 0 2
-3.2 2 1
5 4 7

```

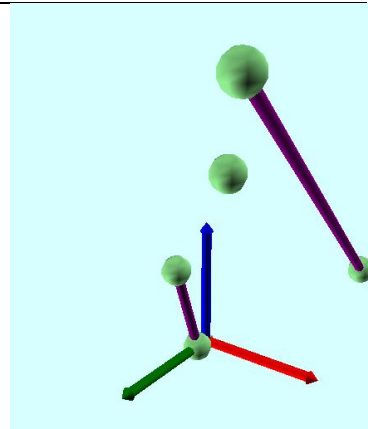


Рис. 6. Пример описания сцены и визуализация по этому описанию

Рассмотрим другой пример, уже из реальной задачи [5]. Дан файл, содержащий набор точек, которые необходимо отобразить с помощью сфер. Тогда решение будет выглядеть следующим образом.

```
1 1 1  
1 2 1  
.....
```

Рис. 7. Выдержка из файла данных data.txt

```
set scene.model.spheres = array * 3 <<file data.txt  
set scene.model.spheres.radius = 0.05  
set scene.model.spheres.color = green
```

Рис. 8. Описание сцены

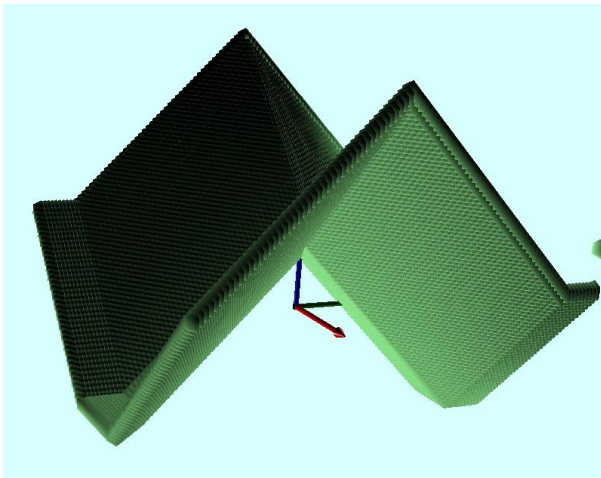


Рис. 9. Визуализация согласно приведенным данным и описанию

Обращаем внимание читателя на краткость, с помощью которой по исходным данным получено визуальное представление. В данном случае оно заняло три строки (рис. 8).

Заключение

В работе представлен метод формирования сцен трехмерной визуализации, направленный на визуализацию результатов численного моделирования.

Среди преимуществ метода авторы видят:

- упрощенный механизм формирования сцены визуализации по сравнению с созданием систем визуализации «с нуля» или написанием модулей для систем визуализации;
- возможность вызова подпрограмм для загрузки и преобразования данных, написанных пользователем на любом удобном ему языке программирования;
- работу с различными системами визуализации, включая веб-ориентированные;
- возможность работы системы визуализации и сервиса данных на разных машинах в сети. Это позволяет передавать не все данные на машину пользователя, а лишь необходимые.

Более подробную информацию по представленному методу визуализации авторы размещают на сайте: www.view.lact.ru.

1. Авербух В.Л., Бараковских Н.И., Зенков А.И., Петров А.Н., Языки описания видов отображения для систем компьютерной визуализации // 14-я Международная Конференция по Компьютерной Графике и Зрению ГрафиКон'2004. 6-10 Сентября 2004, Москва, Россия. Труды Конференции. МГУ им. М.В. Ломоносова. С. 302-305.
2. Васёв П.А, Кумков С.С., Шмаков Е.Ю., О создании среды разработки систем научной визуализации // Труды XIII Международного семинара «Супервычисления и математическое моделирование» (3–7 октября 2011 г.) под редакцией Р.М. Шагалиева. — ИПК ФГУП «РФЯЦ-ВНИИЭФ», г. Саров. С. 131-140.
3. Васёв П.А., Кумков С.С., Шмаков Е.Ю., Конструктор специализированных систем визуализации // Научная визуализация. 2012. Т.4, №2. С. 64–77.
4. Правдин С.Ф., Панфилов А.В., Бердышев В.И., Кацнельсон Л.Б., Соловьёва О.Э., Мархасин В.С. Математическая модель миокарда левого желудочка сердца человека: верификация и расчёт электрического возбуждения // Тезисы докладов Международной (43-й Всероссийской) молодежной школы-конференции “Современные проблемы математики”. Екатеринбург: ИММ УрО РАН, 2012. С. 383–385.
5. Крупенников Е.А., Токманцев Т.Б., Субботина Н.Н. Численное решение задачи идентификации макроэкономической модели // Соврем. проблемы математики: тез. Междунар. 43-й Всерос. мол. шк.-конф.- ИММ УрО РАН, 2012. С. 145-147.

ПОСТРОЕНИЕ ЭФФЕКТИВНОЙ ИТ-ИНФРАСТРУКТУРЫ ДЛЯ ИНЖЕНЕРНЫХ РАСЧЕТОВ

*ОАО «РКЦ «Прогресс», г. Самара,
d883@samspace.ru*

ОАО «РКЦ «Прогресс» – ведущее российское предприятие по разработке, производству и эксплуатации ракет-носителей среднего класса и автоматических космических аппаратов.

В настоящее время промышленные предприятия и инженерные компании могут использовать высокопроизводительные вычисления (High-Performance Computing, HPC) не только для ускорения инженерных расчетов, но и для достижения наиболее глубокого понимания физических процессов, параметрической оптимизации и исследования робастности решения в условиях функционирования изделия в реальном мире. Все чаще ведущие компании полагаются на технологии высокопроизводительных вычислений для достижения максимальной эффективности в процессе проектирования и анализа своей продукции. Однако простое ускорение расчетов не является достаточным. В равной степени важным является создание ИТ-инфраструктуры, которая минимизирует накладные расходы на вычисления и улучшит производительность работы, совместное взаимодействие, хранение и защиту данных на предприятии.

Что такое высокопроизводительные вычисления, насколько данное направление развито в РКЦ и какие перспективы сейчас открываются – излагается в докладе.

Главная тенденция в мире проектирования — это системное моделирование, подход к разработке изделия как к единому целому. Ранее рассматривались подетальная разработка и последующая сборка. Но современные условия требуют рассмотрения изделия как системы. Вводится еще и новый термин — робастность проектирования. Робастность — это устойчивость к колебаниям расчетных/конструкторских параметров. Примером робастной задачи является разработка пассажирского и военного самолета. В зависимости от режима работы надо учитывать все возможные воздействия с их вариациями во время эксплуатации изделия. И это, как следствие, очень повышает нагрузку на современное оборудование. Итак, современная инженерия стремится к работе под девизом олимпийцев — «Быстрее, выше, сильнее».

Цель: разработка и внедрение информационных технологий виртуального проектирования и компьютерного моделирования на базе суперкомпьютеров и кластерной системы для создания и отработки новых

образцов ракетно-космической техники в целях повышения их надежности, качества и технико-экономической эффективности.

Задачи:

- Создание специализированной технологии для расчета характерных процессов, происходящих в ракетах космического назначения (РКН) и космических аппаратах (КА).
- Разработка имитационных 3D-моделей отдельных подсистем ракетно-космической техники.
- Оснащение РКЦ суперкомпьютерами с высокой производительностью.
- Развитие кластерной системы.
- Ввод в опытную эксплуатацию специализированного программного комплекса на основе суперкомпьютеров для расчетов под задачи проектирования РКН и КА различного назначения.
- Решение практически важных задач снижения материалоемкости космических аппаратов и повышения надежности РН.

Вычислительные мощности РКЦ:

- Кластер IBM.
- Супер-ЭВМ.
- Облако на суперкомпьютере «Сергей Королев».
- Облако на ресурсах ФГУП «РФЯЦ ВНИИЭФ».

Компактная суперЭВМ. В соответствии с проектом «Информационные технологии с использованием суперЭВМ для отработки новых образцов ракетно-космических систем» на РКЦ была осуществлена поставка КС-ЭВМ со следующими характеристиками:

- теоретическая пиковая производительность – 1.1 Тфлопс;
- количество вычислительных ядер архитектуры – 144;
- рабочая частота процессоров – 1.9 ГГц.

Системное программное обеспечение данного программно-аппаратного комплекса включает в себя:

- операционную среду параллельных вычислений Scientific Linux SL v5.4;
- коммуникационное программное обеспечение OFED v 1.5.1, OpenMPI v. 1.4.1; инструментальные средства разработки и отладки параллельных приложений;
- программные средства управления доступом и хранением данных (на базе файловой системы Lustre);
- систему мониторинга и диагностики аппаратно-программных компонентов;
- систему управления заданиями;

- оптимизированные прикладные математические библиотеки; параллельные системы визуализации и графического анализа данных для задач математического моделирования;
- систему сбора и анализа информации об эффективности выполнения параллельных приложений;
- пакет программных тестов;
- прикладное ПО «ЛОГОС».

Кластер IBM является высокопроизводительной вычислительной системой, которая интегрирована в локальную сеть предприятия и обладает системой защиты от перепадов напряжения.

Системное программное обеспечение данного программно-аппаратного комплекса включает в себя:

- операционную среду параллельных вычислений Microsoft HPC Windows Server 2008;
- инструментальные средства разработки и отладки параллельных приложений;
- программные средства управления доступом, систему мониторинга, систему управления заданиями Microsoft HPC Job Manager;
- оптимизированные прикладные математические библиотеки;
- коммуникационное программное обеспечение OpenMPI, IntelMPI;
- лицензионные программные продукты Ansys и MSC MasterKey.

В мае 2014 г. управление информационных технологий РКЦ «Прогресс» произвело наращивание мощности кластера IBM в 8 раз с 0,4 до 3,2 ТФлопс. Это не предел, в планах создание гибридного кластера с использованием графических ускорителей и выход на показатель мощности в 10 ТФлопс.

Вычисление в облачных технологиях.

Сложные вычисления ранее производились на больших мейнфреймах, которые занимали целые комнаты. Сейчас развитие данных систем пришло к облачным технологиям. Высокопроизводительная инфраструктура и системы хранения данных централизуются и объединяются для удаленного использования с любых устройств. Зачем все это нужно, спросите вы? Для предоставления высокопроизводительных ресурсов по запросу, для выполнения достоверных расчетов, операционной эффективности и оптимизации активов, для обеспечения совместной работы географически разбросанных команд специалистов и соответственно, для защиты интеллектуальной собственности и контроля процесса работы.

РКЦ «Прогресс» имеет терминальный доступ к вычислительным ресурсам суперкомпьютера «Сергей Королев», СГАУ, 25 ТФлопс, а также к гражданскому сегменту вычислительного центра гражданского пользования «РФЯЦ ВНИИЭФ», г. Саров, 300 ТФлопс.

Описанные выше суперкомпьютерные технологии позволяют решать задачи РКЦ «Прогресс» в следующих областях:

- Прочность.
- Теплофизика.
- Гидрогазодинамика.
- Электромагнетизм.

Результаты проведенных расчетов с хорошей точностью совпадают как с результатами натурных испытаний и экспериментов, так и с результатами расчетов по инженерным методикам.

В результате внедрения суперкомпьютерных технологий с реализованы следующие цели и задачи:

- произведено оснащение подразделений предприятия вычислительной техникой;
- произведено оснащение подразделений предприятия программными комплексами для создания имитационных моделей и ведения сложных инженерных расчетов;
- выполнен анализ существующих информационных систем инженерного анализа, проблем их использования в ракетно-космической отрасли и показана необходимость инновационных подходов к внедрению информационных систем имитационного моделирования в практику создания новых образцов РКТ;
- определены основные направления применения суперкомпьютерных технологий при создании изделий РКТ по Федеральной космической программе России;
- сформулированы ключевые проблемы, связанные с проектированием и определён круг задач, при решении которых необходимо широкое внедрение компьютерного моделирования;
- предложена структура информационного обеспечения этапов проектирования и испытаний при создании изделий РКТ;
- постоянно проводится обучение пользователей.

**ПРОБЛЕМЫ И МЕТОДЫ ОБРАБОТКИ ИНТЕНСИВНЫХ
ПОТОКОВ ДАННЫХ В ПЕРСПЕКТИВНЫХ НАЦИОНАЛЬНЫХ
СИСТЕМАХ МОНИТОРИНГА СЛОЖНЫХ ПРОЦЕССОВ
И МНОГОПАРАМЕТРИЧЕСКИХ СРЕД**

¹ОАО «РТИ», г. Москва,

²Московский физико-технический институт, г. Долгопрудный,

³Институт проблем управления им. В.А. Трапезникова РАН, г. Москва,

⁴Финансовый университет при Правительстве РФ, г. Москва,
baytin@satissoft.ru, dstupin@oaorti.ru, akochkar@gmail.com

Современные и перспективные информационные системы, призванные решать задачи комплексного мониторинга окружающей обстановки, сложных многопараметрических процессов и многомерных физических сред, развиваются в направлении глобализации, т.е. необходимости вовлечения в процесс мониторинга все большего числа различных факторов и временных рядов, а также проведения над этой информацией многофакторного анализа. Это происходит вследствие глобализации и взаимной корреляции практически всех значимых процессов, происходящих в Природе, независимо от причин, породивших эти процессы.

Объем информации, перерабатываемой такими системами и циркулирующей внутри них, оказывается весьма значительным вследствие многопараметрического характера наблюдаемых и анализируемых процессов и огромного числа сенсоров и датчиков, которые являются реальными или потенциальными источниками информации для мониторинговых систем. Наличие многообразных корреляционных связей между параметрами и данными анализируемых процессов, физических сред и т.д. увеличивает объем обрабатываемой информации и усложняет анализ таких систем. Например, в системах мониторинга гидрометеоусловий или геофизического мониторинга атмосферы объемы информации могут составлять десятки петабайт, а потоки – до миллионов условных событий в секунду [1]. При этом во многих системах специального назначения, результатом работы которых являются определенные действия (реагирование), обработка информации для последующего принятия решения должна происходить в реальном времени.

Масштабные и глобальные системы мониторинга прежних поколений (многие из которых функционируют и сегодня) решали проблемы обработки больших массивов и потоков данных (в том числе, в условиях недостаточной производительности вычислительных средств) путем введения этапности обработки, которая позволяла последовательно уменьшать объем обрабатываемой информации. При этом каждый последующий этап об-

работки использовал результаты предыдущего как некое начальное приближение и, возможно, дополнительные данные. Такая процедура позволяла сохранять объем обрабатываемых данных на уровне, доступном для обработки имеющимися средствами и алгоритмами, при сохранении разумных показателей качества «выходной» информации. Однако в этом случае приходилось мириться с неизбежными потерями информации на каждом этапе обработки. Полноценная обработка данных об анализируемой ситуации, при которой потери информации отсутствуют или минимальны, производимая в реальном времени, требует принципиально иных подходов.

Современные мониторинговые системы ориентированы, главным образом, на обработку цифровых данных, то есть информации, имеющей достаточно четкую структуру. Это обеспечивает возможность адекватной структуризации и формализации процессов обработки информации, что, в конечном счете, позволяет создавать аппаратно-программные решения, значительно ускоряющие процесс обработки (например, спецвычислители цифровой обработки или «быстрые» алгоритмы типа БПФ). Однако и в этом случае возможны нарушения структуризации, связанные с аномальными измерениями, изменением характера и характеристик корреляционных связей между измеряемыми параметрами и т.д. Отдельно надо отметить, что при возрастании объема обрабатываемой информации размерности задач, решаемых при этой обработке, могут возрастать многократно, например из-за необходимости учета большого числа новых (скорее всего, многомерных) факторов, влияющих на обстановку или процессы, подлежащие мониторингу. Таким образом, даже при работе со структурированной информацией возможны ситуации, при которых эта структурированность теряется с соответствующими потерями эффективности обработки информации (зачастую именно эти ситуации наиболее «интересны» для применения).

Глобализация социально-экономических, техногенных, природных и т.д. процессов, их усиливающаяся корреляция и взаимосвязь, постоянно увеличивающееся количество разнообразных информационных систем и развитие информационно-телекоммуникационных технологий привели к тому, что перед разработчиками перспективных информационных центров встают следующие серьезные «вызовы»:

- Резко возросли количество, номенклатура и принципы работы разнообразных датчиков и сенсоров, необходимых для адекватного представления анализируемого объекта, а также возможности телекоммуникационной среды по передаче информации от этих датчиков и сенсоров на комплексы обработки информации.
- Для задач комплексного мониторинга существенно возросла роль информации, формируемой различными социумами и пе-

редаваемой через глобальное информационное пространство (ГИП), которая по своей природе является неструктурируемой.

Информация, получаемая от всего многообразия датчиков и сенсоров, как правило, не сводится к единому (или, по крайней мере, унифицированному) формату, собирается, хранится и обрабатывается в различных системах, что затрудняет, а часто и делает невозможным совместную обработку этой информации в реальном времени.

Таким образом, развитие информационно-телекоммуникационных технологий с необходимостью приводит к формированию унифицированных подходов к обработке различных видов информации в системах комплексного мониторинга. Проблема обработки неструктурированной информации в перспективных системах комплексного мониторинга обрела не меньшую (возможно, и большую) значимость по сравнению с задачами обработки структурированной информации. Однако, вне зависимости от степени структурированности обрабатываемой информации, необходимость ее обработки в реальном (в худшем случае – в квазиреальном) времени, а также однотипность критериев оценок ситуации, производимых по результатам обработки, заставляют разработчиков систем мониторинга искать унифицированные решения по построению систем обработки информации различной степени структуризации.

Типовая архитектура системы мониторинга любого масштаба представляет собой комплекс аппаратно-программных средств и систем, в состав которого входят следующие обязательные элементы:

- Совокупность сенсоров и датчиков разного масштаба, предназначенных для непосредственных измерений параметров анализируемой среды (объекта или процесса) и преобразования полученных измерений в информацию, пригодную для трансляции в системы обработки (источники информации из ГИП в этом случае рассматриваются как «варианты» датчиков).
- Вычислительная инфраструктура, предназначенная для обработки информации в соответствии с требованиями потребителей, представления результатов обработки в формате, обеспечивающем адекватную оценку анализируемой ситуации и, возможно, принятие необходимых решений, а также визуализацию и регистрацию результатов обработки.
- Управляющая вычислительная инфраструктура (как правило, сетевая), обеспечивающая контроль и поддержание работоспособности системы и ее отдельных элементов, выбор рабочих конфигураций элементов, а также управление функционированием системы (во многих случаях управляющая вычислительная инфраструктура аппаратно-программно и функционально сопряжена с инфраструктурой обработки информации).

- Информационно-телекоммуникационная инфраструктура, обеспечивающая обмен всеми видами и потоками информации, курсирующей в системе, выбор соответствующих телекоммуникационных каналов и интерфейсов и управление всеми процессами приема-передачи и сбора информации.

Возможность унификации архитектуры системы позволяет разработчику, по крайней мере, пытаться построить унифицированные процедуры обработки информации в такой системе.

Работа с большими и сверхбольшими массивами и потоками информации (включая в этот процесс ее передачу, хранение и анализ), учитывая необходимость работы в реальном времени, требует новых «инструментов» и технологий обработки. Для масштабных (как правило, национального уровня) систем комплексного мониторинга наиболее перспективными представляются технологии «Big Data» [2] и «Big Data Analytics» [3], принципиально ориентированные на работу со сверхбольшими массивами и потоками данных в рамках общего направления «Data Science». В долгосрочном прогнозе научно-технологического развития России до 2030 года [4] эти технологии обозначены среди наиболее приоритетных.

Основные требования к инфраструктуре обработки информации – это масштабируемость (возможность добавления дополнительных серверов для увеличения пропускной способности с минимальными усилиями) и высокая доступность (возможность для достижения непрерывной работы без вмешательства человека при наличии системных сбоев оборудования и сети). Аппаратно-программная реализация этой инфраструктуры должна изначально проектироваться с учетом требований масштабируемости, отказоустойчивости и высокой доступности и, по-видимому, она должна использовать «облачные» технологии.

Технологии Big Data успешно применяются в таких важнейших сферах человеческой жизнедеятельности как медицина, национальная безопасность, автомобильная промышленность, финансы, страхование и т.д. [5]. Они позволяют решать задачи, которые недоступны современным Enterprise-решениям и суперкомпьютерам. Эти технологии позволяют по-новому подходить к созданию ключевого компонента перспективных ситуационных центров и вычислительной инфраструктуры. Поэтому весьма актуальной представляется задача создания на базе технологий «Big Data» единой интеграционной аналитической платформы [6, 7], обеспечивающей сбор, обработку, структуризацию, хранение и анализ больших объемов информации, поступающей от разнообразных сенсоров различной природы и внешних систем. Такая платформа будет обладать следующими нефункциональными характеристиками:

- единой технологической, логической и операционной базой, позволяющей оперативно изменять конфигурацию (добавлять но-

вые модули, изменять структуру имеющихся модулей) в зависимости от поставленной задачи. Это позволит оптимизировать затраты материальных, человеческих и временных ресурсов;

- одновременной обработкой большого количества событий в режиме реального времени;
- возможностью одновременного анализа различных по своей сути источников данных, комбинированием полученных результатов для получения более точной информации в режиме реального времени;
- построением комплексных прогнозных моделей за счет применения большого количества алгоритмов и возможностей одновременной обработки и анализа;
- интеграцией (быстрым предоставлением информации заинтересованным лицам, возможностью оперативной выдачи решения под индивидуальные потребности).

Такая платформа должна быть инвариантна к архитектуре вычислительных средств и программно-алгоритмическому обеспечению, непосредственно реализующим обработку. Необходимость «увязки» в единой системе обработки как структурированной, так и неструктурированной информации приводит к необходимости оптимизации архитектуры вычислительных средств. Представляется наиболее перспективным использование реконфигурируемых вычислительных структур с широкими возможностями по управлению вычислительным ресурсом, включая автоматическое распараллеливание процедур обработки.

Для проектов специального назначения, в особенности связанных с задачами национальной безопасности, такого рода платформы и принципиальные технологии, закладываемые в них, не могут быть «покупными» (вне зависимости от статуса компании-поставщика), а должны разрабатываться исполнителями проекта.

1. Система освещения обстановки в Арктике. Аванпроект. – М.: ОАО «РТИ», 2012.
2. Manyika J, Chui M, Brown B, Bughin J, Dobbs R, Roxburgh C, Byers. Big data: the next frontier for innovation, competition, and productivity. McKinsey Global Institute, 2011.
3. Russom Ph. Big Data Analytics. TDWI Best Practices Report, 2011.
4. Долгосрочный прогноз научно-технологического развития России – 2030. – М.: НИУ «Высшая Школа Экономики», 2013.
5. Большие Данные (Big Data). Электронный ресурс. URL: [www.tadviser.ru/index.php/Статья:Большие_данные_\(Big_Data\)](http://www.tadviser.ru/index.php/Статья:Большие_данные_(Big_Data)).
6. Malfara D. Platform as a Service. ACC 626, 2013.
7. Charu C. Managing and Mining Sensor Data. Aggarwal, Springer, 2013.

ИСПОЛЬЗОВАНИЕ СУПЕРКОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ ДЛЯ МОДЕЛИРОВАНИЯ ПРОЦЕССОВ В ФИЗИКЕ ПОЛУПРОВОДНИКОВ

*Белорусский государственный университет, г. Минск,
bouza@bsu.by*

Методы молекулярной динамики и Монте-Карло используются для моделирования различных процессов в современной физике. Они относятся к классу больших задач, которые требуют для своего решения применения мощных вычислительных ресурсов.

В данном докладе представлены результаты моделирования в физике полупроводников на суперкомпьютере СКИФ-1000-2 и многоядерном персональном компьютере.

Метод молекулярной динамики

Метод молекулярной динамики (МД) применяется в данной работе при моделировании следующих процессов:

- диффузия: система находится при постоянной температуре, отслеживается самодиффузия атомов решетки; материалы – кремний, карбид кремния; используемые потенциалы – SW;
- отжиг: система находится при постоянной высокой температуре, отслеживаются диффузия и рекомбинация дефектов решетки; материалы – кремний; используемые потенциалы – EDIP, SW;
- аморфизация: облучение системы низкоэнергетическими частицами до перехода в аморфное состояние; материалы – кремний; используемые потенциалы – EDIP, SW.

Для моделирования был использован пакет молекулярной динамики XMD (Molecular Dynamics for Metals and Ceramics) [1]. Достоинства пакета: широкий набор поддерживаемых потенциалов, сравнительная простота использования и открытость исходных кодов.

Алгоритм МД может быть представлен как последовательность следующих шагов:

- 1) инициализация координат и скоростей всех атомов;
- 2) построение списка соседей для каждого атома;
- 3) вычисление потенциальной энергии системы и сил, действующих на атомы, используя межатомный потенциал;
- 4) нахождение новых положений и скоростей всех атомов на основе вычисленных сил через некоторый малый временной интервал.

Шаги 2–4 повторяются в течение всего времени моделирования, позволяя найти положения и скорости атомов на каждом временном шаге.

Алгоритм МД обладает естественным параллелизмом: расчет потенциальной энергии системы, сил, координат и скоростей можно выполнять одновременно для всех атомов в пределах одного временного шага. Поэтому вычисления, соответствующие шагу 3 алгоритма МД, могут быть разделены на независимые части.

Метод Монте-Карло

В работе [2] была предложена версия кинетического метода Монте-Карло (КМК) для моделирования эволюции дефектов (собственных междоузлий и их малых кластеров) в кристалле кремния. В основе – решеточный вариант алгоритма КМК, однако при описании малых кластеров их геометрия не учитывалась и выбиралась конфигурация кластера с наименьшей энергией формирования.

В литературе описаны различные подходы к построению последовательных версий алгоритма КМК (решеточный, объектный, событийный и др.). В сети имеются доступные программные реализации этого алгоритма, но каждый из них разработан и оптимизирован для решения определенного класса задач (например, миграция атома по поверхности, изучение кинетики роста пленок и т.п.). Подробная документация для этих программных реализаций отсутствует, некоторые части кода остаются непонятными и играют роль «черных ящиков». Следовательно, практика использования готовых пакетов в данном случае не оправдана. Как правило, каждый исследователь (группа исследователей) создает собственную реализацию алгоритма в зависимости от решаемых задач и вычислительной базы.

В связи с этим авторами доклада создана собственная версия алгоритма для моделирования эволюции дефектов в кристалле кремния.

Алгоритм КМК является по своей сути последовательным: время для моделируемого объекта определяется событиями, на одном шаге по времени происходит ровно одно событие, при этом оно может влиять на последующие шаги. Тем не менее, предлагаются и исследуются различные подходы к разработке параллельных версий алгоритма.

Разномасштабное моделирование

В работе [3] для моделирования начальных стадий формирования кластеров собственных дефектов в кремнии применялся комбинированный алгоритм, в котором для описания миграции одиночных дефектов используется кинетический метод Монте-Карло, а для моделирования процессов роста/распада кластеров – метод молекулярной динамики.

Моделирование, основанное на сочетании методов молекулярной динамики и кинетического Монте-Карло, может быть применено для симуляции ряда процессов, характерных для прикладных вопросов физики полупроводников.

Рассмотрим комбинированный алгоритм КМК+МД на примере формирования кластеров из одиночных вакансий. Будем использовать КМК для описания термически стимулированной миграции вакансий по кристаллу, считая, что параметры этого процесса известны заранее и определены методами МД или квантовой химии. Процессы формирования и распада вакансионных кластеров будем моделировать с использованием классической молекулярной динамики. Причем после появления первого кластера происходит циклическое чередование стадии КМК и стадии МД для исследования роста данного сложного дефекта. Используя рассмотренный алгоритм, мы получаем возможность следить за процессами эволюции (роста/распада) сложных дефектных комплексов. Основным преимуществом данного комбинированного алгоритма является ускорение процесса моделирования кластеров по сравнению с методами МД, поскольку поведение элементарных дефектов моделируется быстрым методом КМК, а эволюция сложного дефекта (кластера) описывается в рамках метода МД, но в специально выбранной области с сокращенным числом атомов.

Для рассмотренных методов построены параллельные алгоритмы и программы и исследована их эффективность.

Параллельные программы с общей памятью

Многопоточность, поддерживаемая современными операционными системами, позволяет создавать параллельные программы с общей памятью. Как правило, технологии разработки программ в этом случае проще, а накладные расходы меньше по сравнению с использованием распределенной памяти.

Преобразовать однопоточную программу в многопоточную гораздо сложнее, чем это может показаться на первый взгляд. Некоторые из возможных трудностей рассмотрены в [4]. Но многопоточные программы отличаются легкостью переноса из-под других операционных систем. Последнее очень важно для программистов, работающих в среде Windows. Поэтому предлагается следующая технология:

- разработать многопоточную программу в Windows-окружении, отладить ее, провести требуемые эксперименты, используя встроенные потоки Windows или POSIX-потоки;
- получить аналог для Unix-системы.

Если на первом шаге была получена реализация на базе POSIX-потоков, то она без изменений может быть использована на суперкомпьютере.

Функционал пакета XMD расширен нами за счет включения многопоточных функций вычисления эмпирических потенциалов для полупроводниковых ковалентных материалов [5]. Для распараллеливания алгоритма МД использовался его естественный параллелизм. Был применен следующий подход к распараллеливанию алгоритма МД: на каждом временном шаге массив частиц разбивался на равные части и вычисление поля сил проводилось независимо на каждом вычислительном узле. Рассматривались задачи: 1 – диффузия, 2 – отжиг, 3 – аморфизация. Некоторые результаты экспериментов показаны в таблице 1.

Результаты экспериментов для метода МД

Таблица 1

Число частиц	2 потока: общее ускорение- ускорение на участке			4 потока: общее ускорение- ускорение на участке		
	1	2	3	1	2	3
4096	1,48-2,23	1,88-2,12	1,21-1,29	2,04-3,89	2,19-3,50	1,43-2,50
8192	1,44-1,94	1,56-2,12	1,20-1,63	1,95-3,05	2,04-3,00	1,33-2,73
16384	1,31-3,56	1,32-1,88	1,10-1,77	1,56-3,94	1,75-4,76	1,15-3,51
32768	1,19-2,10	1,22-1,97	1,05-1,86	1,31-4,10	1,40-3,66	1,08-3,89
65537	1,16-1,92	1,23-1,89	1,03-1,72	1,28-3,80	1,37-3,59	1,09-3,69
Среднее	1,32-2,35	1,44-1,99	1,12-1,65	1,62-3,76	1,75-3,70	1,22-3,27

Для распараллеливания алгоритма КМК использовалась декомпозиция данных, а именно, пространственная декомпозиция моделируемого образца на подобласти (блоки). При использовании данного подхода возникают две основные проблемы: корректное описание переходов частиц через границы подобластей и синхронизация времени в разных блоках (время в каждом блоке определяется происходящими в нем событиями и идет независимо от других блоков). В рамках данного алгоритма рассматривалась задача моделирования диффузионного перераспределения точечных дефектов (вакансий) в кристалле кремния: 1 – при концентрации дефектов 0,02%, 2 – при концентрации дефектов 0,1%. Ряд результатов эксперимента приведен в таблице 2.

Комбинированный алгоритм МД+КМК был применен для моделирования роста кластеров из одиночных дефектов (вакансий или междоузлий) во время отжига ионно-облученного кремния. Данная задача рассматривалась при начальной концентрации дефектов 0,2% в двух случаях: 1 – рост вакансионных кластеров, 2 – рост междоузельных кластеров. Некоторые результаты экспериментов показаны в таблице 3.

Результаты экспериментов для метода КМК

Таблица 2

Объем образца, A^3	2 потока: общее ускорение		4 потока: общее ускорение	
	1	2	1	2
1,001e+8	1,34	1,75	1,61	2,91
2,002e+8	1,61	1,84	2,43	3,08
4,005e+8	1,71	1,89	2,81	3,39
8,010e+8	1,73	1,91	2,90	3,48
Среднее	1,60	1,85	2,44	3,22

Результаты экспериментов для метода МД+КМК

Таблица 3

Размер образца, A^3	2 потока: общее ускорение		4 потока: общее ускорение	
	1	2	1	2
1,001e+8	1,54	1,57	2,00	2,13
2,002e+8	1,57	1,58	2,08	2,14
4,005e+8	1,58	1,59	2,12	2,18
8,010e+8	1,58	1,60	2,16	2,18
Среднее	1,56	1,58	2,08	2,13

1. XMD – Molecular Dynamics for Metals and Ceramics // [Electronic resource]. – March, 2014. – Mode of access : <http://xmd.sourceforge.net/about.html>. – Date of access : 28.04.2014.
2. Моделирование формирования кластеров собственных дефектов в облученном кремнии: кинетический метод Монте-Карло / В.И. Белько [и др.] // III Международная научная конференция «Материалы и структуры современной электроники», БГУ, г. Минск, 25-26 сентября 2008 года. – Минск, 2008. – С. 145-149.
3. Белько, В. И. Моделирование начальных стадий формирования кластеров собственных дефектов в кристалле кремния / В. И. Белько, В. Е. Гусаков, Н. Н. Дорожкин // V Международная научная конференция «Актуальные проблемы физики твердого тела», НПЦ НАН Беларуси по материаловедению, г. Минск, 18-21 октября 2011 года. – Минск, 2011. – Т. 2. – С. 274-276.
4. Таненбаум, Э. Современные операционные системы / Э. Таненбаум. – СПб.: Питер, 2002. – 1040 с.
5. Буза, М. К. Параллельная реализация метода молекулярной динамики на многоядерном процессоре / М. К. Буза, О. М. Кондратьева // Информатика. – 2011. – №1. – С. 44-51.

ВИРТУАЛЬНЫЙ ПОЛИГОН – ЦЕНТР КОМПЕТЕНЦИИ НА ОСНОВЕ СОВРЕМЕННОЙ ТЕОРИИ КАТАСТРОФ

*Национальный исследовательский университет информационных технологий, механики и оптики, г. Санкт-Петербург,
nechaev@ifmo.mail.ru*

Принципиальным достоинством применения методов современной теории катастроф в виртуальном полигоне (ВП), обеспечивающем мониторинг транспортной и социальной систем (TS-систем), является возможность реализации в мультипроцессорной вычислительной среде, организованной на базе исследований [1] – [5]. Это обеспечивает высокую производительность обработки информации, что особенно важно при организации функционирования ВП в режиме реального времени. В отличие от традиционных вычислительных систем методы теории катастроф открывают возможности интерпретации информации при формализации динамической базы знаний, обеспечивая при этом режим самонастройки вычислительной среды в зависимости от полученных решений. В настоящем исследовании рассмотрено расширение проблемной области и функциональных возможностей виртуального полигона на основе концепции современной теории катастроф (рис.1) [4]. Программно-аппаратный комплекс реализован как сложная интегрированная структура «виртуальный полигон – центр компетенции», обеспечивающая мониторинг TS-систем в режиме реального времени.



Рис. 1. Динамическая модель катастрофы, отображающая взаимодействие компонентов исследуемой среды взаимодействия

Реализация парадигмы высокопроизводительных вычислений осуществляется на основе интеграции технологий «облачной» модели и Grid-системы. Концептуальная модель «облачной» технологии обеспечивает настройку серверов в зависимости от способа «виртуализации» совместно используемой инфраструктуры и многоуровневого управления ресурсами TS-систем. Разрабатываемые «облачные» модели обеспечивает единое информационное пространство размещения данных TS-систем. При этом все функциональные элементы перенесены внутрь «облака» и скрыты в интерфейсах доступных сервисов. Таким образом, используемая «облачная» модель является обоснованным аппаратно-программным решением для построения единого информационного пространства, объединяющего потоки данных в ВП как интегрированном программно-аппаратном комплексе нового поколения. Задачи, решаемые в рамках Grid-систем, по своему характеру оказываются близкими к задачам интерпретации TS-систем на основе методов современной теории катастроф и интеллектуальных технологий. Применение многопроцессорных систем при решении сложных задач анализа и интерпретации информации TS-систем эффективно при наличии такой программной поддержки, которая обеспечивает максимальную загрузку всех компонентов Grid-систем, потенциал применения которых в TS-средах существенно зависит от организации вычислений и качества используемого программного обеспечения.

ВП представляет собой сложный вычислительный комплекс, реализующий многоэтапный итерационный процесс формализации знаний при функционировании TS-систем при различном уровне действующих возмущений. Особенность этого комплекса состоит в интеграции знаний об особенностях взаимодействия динамических объектов (ДО) исследуемой TS-системы. На каждом этапе исследования проводилось согласование на концептуальном, алгоритмическом, информационном и программном уровнях разнородных моделей, описывающих функционирования ДО в сложной динамической среде. Выбор допустимых вариантов решения основывался на сжатии исходного множества альтернатив. Концептуальные основы такого анализа предполагают, что оценки затрат на реализацию решений (расход используемого ресурса) не убывали, а становились все более достоверными по мере сужения множества допустимых альтернатив. При этом учитывается, что процессы управления структурной динамикой ВП имеют многоуровневый, многоэтапный и полифункциональный характер.

Эволюция TS-систем отображается на основе синергетической парадигмы [5]. В рамках такой интерпретации используется подход, реализуемый с использованием *обобщенного принципа диссипации*, сформулированного Н.Н.Моисеевым [3]. Применительно к исследованию

TS-систем этот принцип имеет следующую интерпретацию: *если допустимо не единственное состояние транспортной или социальной среды, а целая совокупность состояний, согласованных с действующими законами и связями, наложенными на TS-систему, то реализуется то состояние, которому отвечает минимальный рост энтропии*. Этот принцип носит универсальный характер и может быть использован при изучении поведения TS-систем.

Процесс эволюции состоит в переходе системы из одного состояния в другое за счет изменения управляющих воздействий. В результате система сама начинает движение в сторону нового состояния, которое она занимает в течение некоторого интервала времени. В синергетике такой переход основан на принципе *бифуркационного управления* рисками. Развитие динамики TS-систем в этом случае осуществляется на основе следующих аксиом, сформулированных в рамках обобщенного принципа диссипации [3].

Аксиома 1. *Бифуркационная диаграмма, отображающая движение TS-системы к целевому аттрактору, представляет собой каскад бифуркаций. При этом движение по регулярным участкам эволюции чередуется с точками бифуркации, характеризующими ситуации выбора «параметра порядка» в самоорганизующейся TS-системе на основе синергетической парадигмы.*

Аксиома 2. *Движение по каскаду бифуркаций в случае потери устойчивости объектов транспортной и социальной среды характеризуется траекторией движения TS-системы, отображающей возрастание опасности наступления критической ситуации (катастрофы).*

Аксиома 3. *Принцип вложения аттракторов TS-системы определяет базовые понятия синергетической парадигмы, которые можно интерпретировать как декомпозицию пространства состояний TS-систем в процессе самоорганизации, приводящую к выделению параметров порядка, к которым подстраиваются остальные параметры системы.*

Реализация аксиом 1 – 3 позволяет выявлять «скрытые» эффекты и явления, возникающие в TS-системах. Вследствие декомпозиции пространства состояний формируется определенная направленность процессов самоорганизации и образуются *целевые аттракторы*, имеющие свои области притяжения в зависимости от особенностей фрактальных множеств, определяющих структуру динамической модели катастроф [4]. Это свойство нелинейных диссипативных систем объясняется эффектом самоорганизации и открывает новые возможности *анализа рисков* принятия решений в TS-системах при интерпретации чрезвычайных ситуаций [4],[5].

Функционирование ВП осуществляется на основе аппаратно-программного комплекса в соответствии со структурой, представленной на рис.2. Здесь использованы следующие обозначения: А – блок моделей взаимодействия объектов TS-систем; В – блок управления и интерпретации; С – блок информационного обеспечения; D – блок интеллектуальных технологий реализации баз знаний TS-систем.

Рассмотрим характеристику блоков ВП. M_1 – блок моделей оценки состояния TS-систем, навигационной и оперативной обстановки; MA_i ($i=1, \dots, n$) – модели анализа поведения TS-систем и внешней среды; M_2 – блок моделей управления; $M\Phi$ – модели функционирования ВП; MU_j ($j=1, \dots, m$) – модели взаимодействия элементов управляющего комплекса; MU_k ($k=1, \dots, K$) – модели взаимодействия элементов TS-систем; M_3 – блок планирования операций; $(MLP)-1, \dots, MLP-m$ – блоки долгосрочного и оперативного планирования элементов ВП; $MOP-1 \dots MOP-n$ – модели долгосрочного и оперативного планирования; $MC-1, \dots, MC-k$ – модели управления структурами компонентов ВП; DS – диалоговая система управления ВП; LS_p ($p=1, \dots, P$) – локальные системы управления; SA – блок генерации сценариев и адаптации TS-систем; RB – блок выработки практических рекомендаций; $IS-1, \dots, IS-m$, $IS-1, \dots, IS-n$ – ИС контроля состояния ВП, ДО и внешней среды в TS-системах; VR – система виртуальной реальности; $SC\&MAC$ – технологии мягких вычислений и мультиагентных систем; $GRID$ – технологии реализации виртуального полигона; $CLOUD$ – технология «облачных» вычислений.

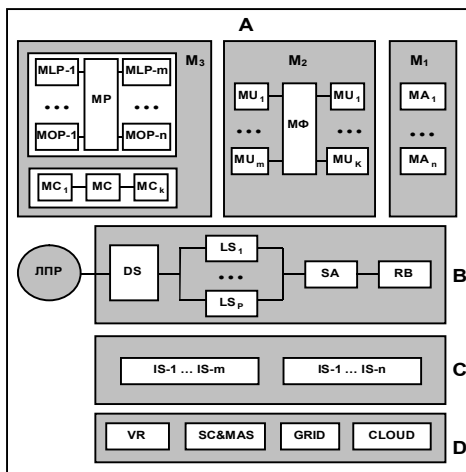


Рис. 2. Структура ВП контроля чрезвычайных ситуаций в TS-системах

Содержательная интерпретация практических приложений ВП рассмотрена применительно к оценке текущих ситуаций в TS-системах. Основное внимание уделяется развитию концепции фрактальных отображений и граф-структур в рамках динамической модели катастроф. Теоретическая база создания ВП формируется на основе эффективного сочетания накопленной системы знаний с новыми подходами и парадигмами искусственного интеллекта. В качестве базовой концепции при разработке программно-аппаратного комплекса использована технология iPSE (intelligent Problem Solving Environment) [1], расширение которой достигнуто в рамках «облачной» модели системы VLUG (Virtual Urgent) [2]. Повышение достоверности оценки и прогноза ситуации достигается с использованием парадигмы обработки информации в мультипроцессорной вычислительной среде [4], основанной на развитии концепции «мягких вычислений». Парадигма предусматривает использование теоретических принципов, позволяющих обеспечить рациональную организацию вычислительной технологии обработки данных измерений в задаче анализа и прогноза развития экстремальной ситуации, а также формализовать поток информации при реализации нечеткого логического вывода в мультипроцессорной вычислительной среде.

Целью управления и принятия решений при функционировании ВП является достижение желаемого (целевого) аттрактора, т.е. асимптотически устойчивого конечного состояния TS-систем. Размерность целевого аттрактора существенно меньше размерности исходного пространства контролируемых сред. Это позволяет сформулировать общий подход к моделированию поведения объектов TS-систем в рамках синергетической парадигмы [5]. Процессы обработки информации в интегрированном вычислительном комплексе ВП включают две фазы — фазу расширения и фазу сжатия пространства состояний ДО транспортной и социальной среды. Эти фазы реализуются путем совокупности управляющих воздействий. В фазе расширения формируется подмножество различных альтернатив поведения транспортной и социальной среды. В фазе сжатия преобразуется область притяжения множества аттракторов в один из желаемых (целевых) аттракторов. Стратегия направленной самоорганизации при функционировании ВП состоит в формировании и поддержании системных инвариантов, определяющих структуру соответствующих аттракторов – целей управления и принятия решений в TS-системах. На основе этих утверждений сформулирована общая задача синтеза ВП контроля TS-систем при управлении и принятии решений на основе синергетического подхода. Формальная модель, определяющая динамику взаимодействия объектов в TS-системах, описана с помощью параметров, определяющих состояния

исследуемой среды, вырабатываемых управляющих внешних и параметрических воздействий.

Как следует из теоретических принципов современной теории катастроф [4], эволюция TS-систем интерпретируется в виде двух предельных случаев взаимодействия. Первый случай определяется областью притяжения, определяющей движение системы к целевому аттрактору, а второй – областью потери устойчивости (возникновение катастрофы). Управляющие воздействия для рассматриваемых случаев эволюции определяют эффективную и недостаточно эффективную интеллектуальную поддержку, когда не удается реализовать вырабатываемые мероприятия по обеспечению безопасности социальной среды.

Таким образом, рассмотренный подход открывает новые возможности анализа и прогноза взаимодействия объектов в TS-системах. Онтологический синтез аксиоматики вычислений позволил формализовать процесс обработки информации и структурировать систему знаний исследуемой среды в рамках нечеткого логического базиса. Циклы эволюции TS-системы в текущей ситуации рассматриваются на основе динамической модели катастроф. В процессе эволюции определяются критические ситуации и интервалы времени возникновения неблагоприятных режимов функционирования. Анализ этих режимов открывает возможности генерации управляющих воздействий по предотвращению чрезвычайных ситуаций. Формирование фрактального аттрактора и генерация соответствующего правила с помощью логической системы знаний обеспечивают оперативный контроль и прогноз развития ситуации в заданных условиях функционирования объектов TS-систем.

1. Бухановский А. В., Васильев В. Н. Современные программные комплексы компьютерного моделирования e-Science // Изв. вузов. Приборостроение. 2010. № 3. С. 60–64.
2. Бухановский А.В., Иванов С.В., Нечаев Ю.И. Моделирование нелинейных систем в сложных динамических средах. – Монография: Санкт-Петербург. Изд. Цент НИУ ИТМО, 2013.
3. Моисеев Н.Н. Избранные труды. М. Тайрекс Ко. 2003.
4. Нечаев Ю.И. Теория катастроф: современный подход при принятии решений. – Санкт-Петербург: Арт-Экспресс, 2011.
5. Системы искусственного интеллекта в интеллектуальных технологиях XXI века. – Санкт-Петербург: Арт-Экспресс, 2011.

А.В. Бухановский, С.В. Иванов, С.В. Ковальчук, Ю.И. Нечаев

**ПРОГРАММНЫЙ КОМПЛЕКС «ВИРТУАЛЬНЫЙ ПОЛИГОН –
ЦЕНТР КОМПЕТЕНЦИИ», ОБЕСПЕЧИВАЮЩИЙ
ОПЕРАТИВНЫЙ КОНТРОЛЬ ЧРЕЗВЫЧАЙНЫХ СИТУАЦИЙ
В ТРАНСПОРТНОЙ И СОЦИАЛЬНОЙ СРЕДЕ НА ОСНОВЕ
ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ И ГРИД-ТЕХНОЛОГИЙ**

*Национальный исследовательский университет информационных
технологий, механики и оптики, г. Санкт-Петербург,
nechaev@ifmo.mail.ru*

Интегрированный программно-аппаратный комплекс «виртуальный полигон – центр компетенции» позволяет осуществлять непрерывный контроль динамики транспортных и социальных систем (TS-систем) при различном состоянии внешней среды. В процессе разработки комплекса использован подход к обработке данных динамических измерений, предусматривающий создание алгоритмов и программного обеспечения встроенных процедур анализа и прогноза рассматриваемой экстремальной ситуации [1]-[5]. Функционирование программного комплекса осуществляется с учетом непрерывного изменения динамики объектов TS-системы на базе вычислительных структур с высокой степенью параллелизма в целях поддержки принятия решений по обеспечению безопасности систем в сложной динамической среде. Пространство состояний TS-системы определяется особенностями ее функционирования на основе динамической модели катастроф [4]. В рамках такой интерпретации поведение TS-системы отображается в виде упорядоченного множества фрактальных структур. Такое отображение представляет собой самоподобные структуры, развивающиеся в процессе эволюции TS-системы. В общем виде фрактальный ансамбль изображается в виде компактного множества топологического пространства, представленного на основе общей модели TS-системы.

Стадии *стабильного* развития определяют движение TS-системы к целевому аттрактору. В этой ситуации система остается в состоянии, близком к равновесному, а ее организация не претерпевает значительных изменений. При стабильном состоянии воздействия внешних возмущений на TS-систему описываются потоковой нагрузкой, изменяющей структуру фрактального графа. Движение к целевому аттрактору при таком состоянии определяется динамической моделью катастроф. Стадия *критического* состояния (возникновение катастрофы) характеризуется выходом исследуемых параметров TS-системы из заданного диапазона и возникновению бифуркаций – потери устойчивости системы, что связано с формированием альтернативных вариантов ее орга-

низации. Такты системы определяют процесс самоорганизации фрактальной структуры в условиях непрерывного изменения динамики объекта и внешней среды.

Внешние возмущения и внутренние противоречия в системе являются источниками неопределенности. Эти особенности порождаются большим числом действующих факторов, из которых выбираются только факторы, существенные для рассматриваемой задачи. При значительных возмущениях фрактальная система должна выполнить сложные преобразования, отображающие эволюцию в условиях самоорганизации. Движение системы при потере устойчивости тесно взаимосвязано с явлением «скачка» при фазовом переходе в новое состояние (возникновение катастрофы). Моделирование процессов самоорганизации в TS-системе осуществляется на основе синергетической парадигмы [5], позволяющей формализовать динамику *фрактального ядра* $F_R(G)$ в соответствии с принципами «расширения – сжатия».

Интегрированная среда интеллектуальной поддержки моделирования и визуализации при функционировании комплекса «виртуальный полигон – центр компетенции», обеспечивающий оперативный контроль динамики TS-систем, содержит компоненты динамической среды и ее характеристик, оценку состояния TS-систем с помощью логических моделей, математическое моделирование текущей ситуации на основании интегрированных вычислительных сред, выработку практических рекомендаций, анализ альтернатив и принятие решений.

Формализованное ядро вычислительного комплекса поддержки процессов управления при контроле динамики TS-систем реализуется в рамках нечеткого логического базиса. Фундаментальной основой такой интерпретации является концепция нечетких целей и ограничений. Повышение эффективности функционирования процедурного компонента интегрированной базы знаний ВП достигается за счет использования принципа конкуренции и формализации процесса обработки нечеткой информации в высокопроизводительной вычислительной среде. Другими принципами повышения эффективности ВП являются принцип открытости, принцип сложности и принцип нелинейной самоорганизации. Реализация этих принципов осуществляется в рамках концепции мягких вычислений [3], интегрирующей нечеткую логику, теорию искусственных нейронных сетей (ИНС) и генетический алгоритм. Методологические основы и формальный аппарат используемых методов и моделей определяют решение задач диагностики и моделирования в сложных динамических средах [1]-[3].

Практическое применение комплекса «виртуальный полигон – центр компетентности» связано с идентификацией, оценкой и прогнозом развития экстремальных ситуаций, возникающих в TS-системах.

Концепция организации комплекса как системы реального времени в рамках принципа конкуренции позволяет обеспечить непрерывный контроль динамики контролируемых систем в сложной динамической среде. В рамках этой концепции эволюция системы определяется внешними условиями и реализуется через самоорганизацию и адаптацию компонентов комплекса к внешним воздействиям. В процессе развития TS-системы меняются ее состояние и сложность, которые определяются накопленной информацией. В результате обеспечивается возможность оперативного контроля и прогноза поведения элементов TS-систем как сложных динамических объектов (ДО) в условиях неопределенности.

Программный комплекс имитационного моделирования поведения объектов TS- систем содержит вычислительные блоки, поддерживающие процедуры анализа и интерпретации исходной и аналитической информации (рис.1). Моделирование процессов развития сложных ситуаций и механизмов управления изменениями функционирования комплекса «виртуальный полигон – центр компетенции» реализуется в рамках концепции современной теории катастроф [4].

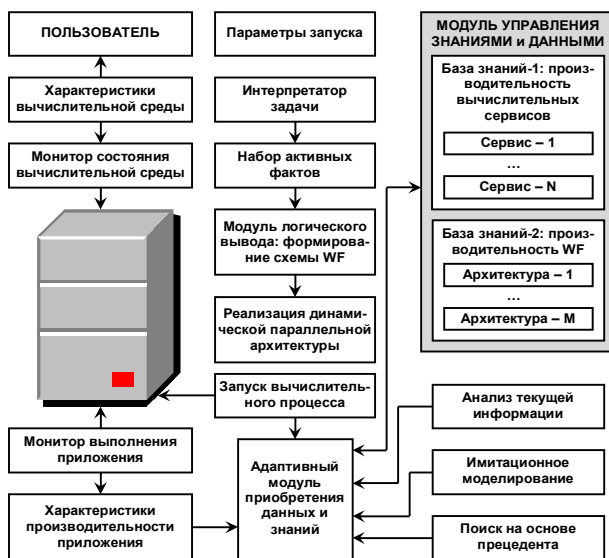


Рис. 1. Структура управления параллельными процессами TS-систем

Монитор состояния вычислительной среды и монитор выполнения приложений передают информацию на сервер, обеспечивающий функционирование программного комплекса в режиме реального времени.

Уровень аппаратной конфигурации и компьютерного оборудования определяет модуль управления знаниями и данными, с помощью которого поддерживается производительность вычислительных серверов и потока данных WF. Интеллектуальная поддержка функционирования комплекса обеспечивается адаптивным модулем, на основе которого осуществляется запуск вычислительного процесса с использованием характеристики производительности приложения и блоков анализа текущей информации, имитационного моделирования и поиска на основе прецедентов. Информация на сервере включает сервисы, оборудование «облака» и характеристики системы: объем оперативной памяти, скорость передачи данных, частоту процессора и объем видеопамати.

Модель «облачных» вычислений реализуется концептуальным блоком на основе многоуровневого интерфейса пользователя, интеллектуальной системы и управления вычислениями с использованием описания задачи, формализации описания и предметной области, *метапотока* данных MWF и блока обработки данных. При этом осуществляется декомпозиция задачи и ее отображение на набор доступных метасервисов с использованием дополнительных требований к процедуре решения. Полученная информация передается на блок *абстрактного потока данных* AWF, технически настроенного потока данных WF, актуализация состояния которого осуществляется от блока управления. Формирование и обработка *конкретного потока данных* CWF ведутся с использованием статуса задачи, а результаты работы передаются в блок анализа и блок полученных результатов решения. В соответствии с организацией ВП описание моделируемых ситуаций в TS-системах осуществляется с помощью интерпретаций на основе сервисно-распределенной архитектуры (рис.2).



Рис. 2. Сервисно-распределенная архитектура организации параллельных вычислений при интерпретации TS-систем

В этой структуре сервисы (A,...,H) реализуют множество задач, возникающих при интерпретации объектов среды: A – внешняя среда; B – динамика объекта; C – динамика взаимодействия (GA-система); D – динамика взаимодействия (ND-система); E – визуализация динамических сцен; F – генерация и анализ альтернатив; G – выработка практических рекомендаций; H – оценка риска принимаемых решений.

Для описания процессов в TS-системах применяются различные математические формализмы: графовые модели, уравнения популяционной динамики, мультиагентные системы и пр. Для отражения характера взаимосвязей между объектами транспортных и социальных систем одновременно на микро- и на макроуровне применяется специальный класс моделей статистической механики – формализм комплексных сетей. Динамика TS-систем развивается в соответствии с конкретными задачами контроля поведения исследуемой среды и формализуется на основе принципа конкуренции [3] – [5] в рамках концепции мягких вычислений [5]. В рамках описанного подхода осуществляется формализация задачи генерации и принятия решений на основе современной теории катастроф [4]. Для этого используется модель преобразования информации, содержащая вектор условий, определяющий множество условий существования решений, вектор решений, включающий множество решений и матрицу соответствия, задающую модель связи между условиями и решениями. Решение проблемы неоднозначности при выборе наилучшего решения осуществляется на основе нечеткой логики. Для входных параметров строятся функции принадлежности и по ним определяются соответствующие степени принадлежности. Для символьных входных параметров значения степеней принадлежности устанавливают субъективно – на основе бинарных сравнений. В сложных задачах анализа и интерпретации экстремальных ситуаций проблема неоднозначности решается с помощью интегрированных нейронных сетей. Разработанная концепция и подход к анализу ситуаций на основе современной теории катастроф осуществляются в ВП на основе интегрированной динамической базы знаний. Такая структура позволяет формализовать поток информации путем различных связей между входными и выходными параметрами исследуемых нестандартных (нештатных и экстремальных) ситуаций. Разрабатываемые программные средства функционирования ВП позволяют использовать достоинства различных формализмов – компактность, простоту и полноту содержания знаний в продукционных моделях, реализуемых в мультипроцессорной вычислительной среде [5].

Приложение интеллектуальных технологий ВП рассмотрено при контроле чрезвычайных ситуаций в морских транспортных системах при контроле динамики судна на волнении в экстремальных ситуациях

при воздействии разрушающихся и экстремальных волн, а в социальной сфере – при контроле наводнений в Санкт-Петербурге.

Заключение

Разработанные теоретические основы интеграции моделей взаимодействия объектов TS-среды на основе концепции динамической модели катастроф открывают возможности контроля текущих ситуаций при поддержке принятия решений. В рамках такой интерпретации реализована логико-математическая структура комплекса «виртуальный полигон – центр компетенции» с помощью формального аппарата знаний. Логическая система знаний мониторинга TS-среды позволяет построить структуру объектов взаимодействия для исследуемой проблемной области в зависимости от особенностей текущей ситуации. Анализ динамики исследуемой среды рассмотрен на базе методов и моделей, отображающих поведение системы с помощью структур знаний на интервале реализации. Встроенные процедуры механизма логического вывода обеспечивают анализ и прогноз ситуаций в TS-средах в рамках концепции «мягких вычислений». Выбор предпочтительной вычислительной технологии при интерпретации текущих ситуаций осуществляется на основе парадигмы обработки информации в мультипроцессорной вычислительной среде.

1. Бухановский А.В., Васильев В.Н., Виноградов В.В., Смирнов Д.Ю., Сухо-руков С.С., Яппаров Т.Г. Перспективная технология «облачных» вычислений второго поколения // Изв. вузов. Приборостроение. 2011. Т.54. №10. С.7 – 15.
2. Васильев В.Н., Князьков К.В., Чуров Т.Н., Насонов Д.А., Марьин С.В., Ковальчук С.В., Бухановский А.В. CLAVIRE: облачная платформа для обработки больших объемов данных // Информационно-измерительные и управляющие системы. 2012.. Т. 10. № 11. С. 7-16.
3. Бухановский А.В., Иванов С.В., Нечаев Ю.И. Моделирование нелинейных систем в сложных динамических средах. – Монография: Санкт-Петербург. Изд. Цент НИУ ИТМО, 2013.
4. Нечаев Ю.И. Теория катастроф: современный подход при принятии решений. – Санкт-Петербург: Арт-Экспресс, 2011.
5. Системы искусственного интеллекта в интеллектуальных технологиях XXI века. – Санкт-Петербург: Арт-Экспресс, 2011.

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ В ПРОИЗВОДСТВЕ ИНТЕГРАЛЬНЫХ МИКРОСХЕМ

*Объединенный институт проблем информатики НАН Беларуси,
г. Минск,
doudkin@newman.bas-net.by*

Введение

Системы компьютерного зрения (СКЗ) являются важной составной частью аппаратно-программного обеспечения производства интегральных схем (ИС). С их помощью решаются задачи проекционного переноса топологии с помощью устройств пошагового совмещения (степперов), контроля топологических слоев, моделирования процесса фотолитографии, покрытия схем элементами экспонирования при подготовке информации для генераторов топологии, преобразования форматов топологических данных и др. Изображения топологических слоев ИС (металлизации, диффузионного или поликремневого и др.) получают с помощью специальных или стандартных устройств ввода данных и представляются множеством растровых кадров, которое формирует поток данных на входе СКЗ. Задача СКЗ – обеспечить обработку потока за минимальное время с применением различных алгоритмов для разных слоев [1-5].

Использование параллельных вычислений при обработке больших объемов данных часто позволяет сократить временные затраты. Однако способ распараллеливания следует выбирать исходя из условий решаемой задачи. Так, использование GPU будет оправданным в случае большого количества однотипных и достаточно сложных вычислений над данными, представленными в виде чисел с плавающей запятой. В случае же более простых вычислений либо при работе с целочисленными данными выгоднее использовать распараллеливание вычислений на отдельные ядра процессора с помощью спецификации OpenMP.

Основные подсистемы обработки

Программное обеспечение СКЗ проектирования ИС содержит следующие основные подсистемы:

- подсистему склейки кадров;
- подсистему подготовки информации для генераторов изображений ИС;
- подсистему конвертации топологических данных из формата внутреннего представления в формат установок автоматического контроля;

- подсистему контроля процессов фотолитографии;
- программный комплекс моделирования процессов фотолитографии для автоматического контроля оригиналов топологии СБИС.

Организация параллельной обработки

Сценарий алгоритма для организации выполнения на кластере представляется на языке XML и содержит описание операций с параметрами вызова и типов кадров, очередь типов операций, а также распределение их по процессорам. Сценарий подается на вход интерпретатора и выполняется для всего множества входных данных. Интерпретатор осуществляет разбор файла сценариев, предоставляет исполнителям необходимую для формирования контекста вызова информацию, управляет созданием новых вычислительных гранул и постановкой их в очередь планирования.

В состав архитектуры входят клиент (на основе платформы Windows) и параллельный сервер (на основе платформы Linux). Взаимодействие между клиентом и сервером осуществляется через файловую систему. Клиент на основе ОС Windows осуществляет выполнение задач, обеспечивающих интерфейс системы с пользователем. Он содержит хранилище данных и ПО системы, позволяющее пользователю выполнить все операции для обработки слоев и склейки кадров. Здесь пользователь может задать параметры, которые будут использованы при работе параллельной системы. Клиент также содержит средства удаленного вызова процедур параллельного сервера.

Клиент имеет доступ к общему каталогу на сервере с помощью средств сервера Samba. Сервер настраивает определенный каталог для общего доступа, и клиент может монтировать его при загрузке Windows. В данный каталог помещаются файлы для обработки (например кадры), а также сценарии обработки. Сценарий представляет собой сгенерированный средствами ПО клиента текстовый файл в синтаксисе Linux shell, содержащий команды запуска процедур параллельного сервера с требуемыми параметрами командной строки, указанными в описании процедур. Сгенерированный сценарий помещается в общий каталог в виде исполняемого командного файла. Выполнение сценария со стороны клиента осуществляется посредством протокола SSH (Secure SHell).

Сервер представляет собой параллельный вычислительный кластер на базе ОС Linux, который использует библиотеку поддержки параллельных вычислений MPICH и настраивается на использование SSH без пароля в качестве команды удаленного запуска.

Вычислительные операции взаимодействуют со специальным механизмом хранения данных, входящим в архитектуру платформы организации параллельных вычислений. Этот механизм реализует интер-

фейс доступа к разделяемой памяти, в которой находятся данные и результаты операций. Механизм доступа обеспечивает запись или чтение переменной с определенным именем, однозначно определяющим ее. Структура вычислительной гранулы и ее взаимодействие с механизмом хранения данных показаны на рис. 1.

Вычислительные гранулы объединяются в специализированные библиотеки, которые динамически подключаются при выполнении параллельного приложения. Гранула загружается из библиотеки при необходимости ее выполнения и идентифицируется по имени операции. Реализация различных классов алгоритмов обработки информации в виде библиотек вычислительных гранул позволяет расширить сферу применения системы организации параллельных вычислений и дает возможность создавать новые классы приложений.

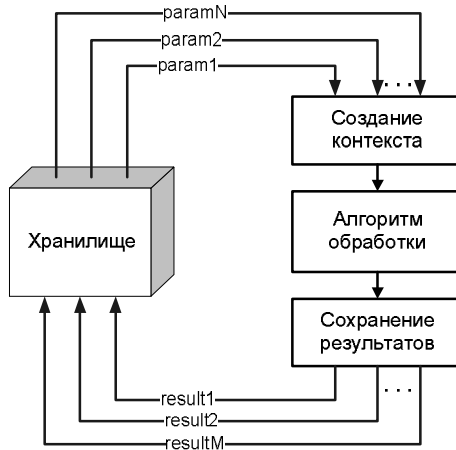


Рис. 1. Взаимодействие гранулы с хранилищем данных

Многоагентная система времени выполнения для организации параллельных вычислений содержит два типа программных агентов: координатор и исполнитель. Агенты реализованы как процессы MPI и используют средства MPI для обмена информацией и управления вычислениями. Количество агентов-исполнителей при выполнении параллельного приложения равно количеству физических процессоров целевой вычислительной системы плюс один агент-координатор. Взаимодействие между агентами осуществляется с помощью передачи сообщений, реализующих простой протокол взаимодействия. Взаимодействие агентов системы времени выполнения представлено на рис. 2.

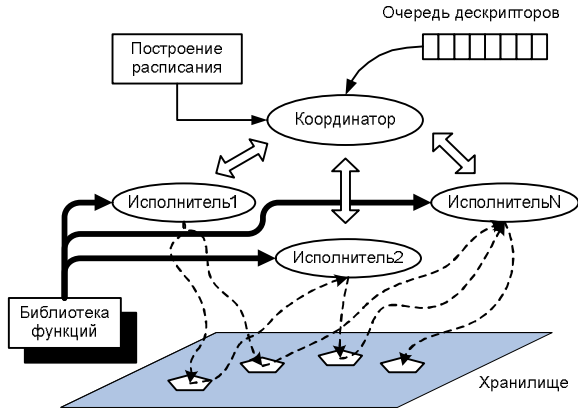


Рис. 2. Архитектура системы времени выполнения

Исследование параллельного приложения

Для оценки разработанной платформы организации параллельных вычислений были проведены две серии экспериментов: для детерминированных потоков, имеющих фиксированное количество объектов с определенной последовательностью типов, и для стохастических потоков, когда исходные объекты и их типы генерировались случайным образом. Эксперименты проведены на суперкомпьютере СКИФ-ОИПИ [6].

Исследование механизма распараллеливания покажем на примере работы подсистемы моделирования фотолитографии. Были проведены эксперименты по измерению производительности в зависимости от количества использованных процессоров, а также в зависимости от исходных данных (размера исследуемого фотошаблона).

Первая группа экспериментов показывает зависимость скорости обработки 50 объектов данных от количества процессоров, выделенных для запуска приложения. Как видно из результатов, общее ускорение обработки составляет примерно три раза для случая четырех процессоров. Этот результат обусловлен тем, что параллельное приложение имеет нелинейную структуру с различной продолжительностью операций, поэтому его параллелизм также нелинеен. Тем не менее, полученное ускорение позволяет существенно повысить скорость обработки фотошаблонов.

Вторая группа экспериментов показывает зависимость производительности параллельного приложения от размера входных данных (количество обрабатываемых объектов фиксировано и равно 100, количество процессоров 4). Алгоритм показывает практически линейную масштабируемость. При увеличении количества точек зрачка в два раза по

координатам X и Y объем данных для обработки увеличивается в четыре раза. При этом время обработки возрастает в 3,16 раза для 100 точек и в 3,42 раза для 200 точек (относительно времени для 100 точек). Это свидетельствует о хорошей масштабируемости алгоритма, поскольку время работы практически линейно зависит от поступающего объема данных.

Заключение

Эксперименты на суперкомпьютере СКИФ показывают, что реализация параллельных приложений в рамках специализированных компонентных архитектур дает возможность пользователям существенно ускорить процессы создания, анализа и оптимизации программ. Архитектура средств обработки потоков данных, основанная на использовании многоагентных систем, может быть легко адаптирована для многих приложений, предполагающих параллельную обработку. Использование компонентного подхода определяет гибкую структуру параллельного приложения, позволяющую адаптировать вычислительный процесс к условиям функционирования. Средства проектирования легко расширяются новыми операциями, алгоритмами и типами данных для реализации приложения обработки информационных потоков из различных предметных областей.

1. Yu, D. Z. Pan and C. A. Mack. Fast lithography simulation under focus variations for OPC and layout optimizations. Proc. SPIE 6156, Apr. 2006, pp. 397–406.
2. Eric R. Poortinga. Comparing software and hardware simulation tools on an embedded-attenuated PSM / Eric R. Poortinga [et al.] [Electronic resource]. – 2007. – Mode of access: www.micromagazine.com/archive/00/06/poortinga.html. – Date of access: 02.07.2012.
3. G. A. Gomba. Collaborative Innovation: IBM's Immersion Lithography Strategy for 65 nm and 45 nm Halfpitch Nodes & Beyond. Proc. SPIE 6521, 2007.
4. O. Beaumont, A. Legrand, Y. Robert. Static scheduling strategies for heterogeneous systems. Computing and Informatics. 21 (2002). p. 413–430.
5. T. Hagras, J. Janecek. A Fast Compile-Time Task Scheduling Heuristic for Homogeneous Computing Environments. International Journal of Computers and Their Applications. 12 (2) (2005). p. 76–82.
6. Принципы создания базовых конфигураций суперкомпьютерных систем отраслевого назначения / В. В. Анищенко [и др.] // Информатика. – 2012. – №1 (33). – С. 97–105.

ПАРАЛЛЕЛЬНАЯ ПОДСИСТЕМА ДЛЯ МОНИТОРИНГА СОСТОЯНИЯ ЛЕСНЫХ УГОДИЙ

*Объединенный институт проблем информатики НАН Беларуси,
г. Минск,
doudkin@newman.bas-net.by*

Введение

В настоящее время эффективным инструментом экологического и сельскохозяйственного мониторинга с целью предотвращения чрезвычайных ситуаций является обработка и анализ мультиспектральных изображений высокого пространственного разрешения лесных и заболоченных районов, получаемых со спутников ДЗЗ. Надлежащая обработка мультиспектральных изображений высокого пространственного разрешения позволяет обнаружить повреждения леса на ранней стадии. Базовой процедурой в тематической обработке мультиспектральных снимков, получаемых со спутников ДЗЗ, является сегментация на основе многомерного кластерного анализа. В процессе сегментации мультиспектральных изображений несколько каналов обрабатываются одновременно, при этом пространственный анализ и многомерный кластерный анализ тесно объединяются в одном алгоритме. Пространство кластерного анализа формируется из каналов исходного изображения. Это позволяет учесть тот факт, что пиксели одного сегмента изображения, помимо всего прочего, расположены компактно в оптическом спектральном пространстве, т.е. спектральные цвета пикселей, принадлежащих одному сегменту изображения, различаются гораздо меньше, чем средние спектральные цвета различных сегментов. Для того чтобы учесть пространственные соотношения между пикселями, это пространство может дополняться другими измерениями, отражающими реальные метрические соотношения между пикселями изображения. Такая конструкция пространства позволяет достаточно хорошо сегментировать области и выделить объекты, в пределах которых спектральные цвета изменяются незначительно.

Структура аппаратно-программного комплекса

АПК мониторинга состояния лесных угодий предназначен для осуществления контроля развития заболеваний растительности, основными компонентами которого являются:

- 1) файловая система / ГИС – предназначена для хранения исходных данных;

- 2) подсистема распараллеливания обработки – для выполнения обработки растровых данных;
- 3) подсистема взаимодействия с пользователем – позволяет осуществлять контроль и управление обработкой;
- 4) подсистема принятия решений – для принятия решения о необходимости внесения химикатов и их количестве, исходя из результатов обработки, данных системы управления внесением химикатов и параметров работы, задаваемых пользователем (выбор алгоритма обработки и их параметров);
- 5) система управления внесением химикатов – для формирования задания системе внесения химикатов; содержит данные, которые в зависимости от плотности заболеваемости растений, позволяют приготовить раствор химикатов требуемой концентрации.

Предлагается методика осуществления мониторинга в разработанном АПК, которая включает:

- формирование информативных признаков и подбор значений их параметров;
- обучение нейросетевого классификатора системы с использованием экспертных данных о здоровых и пораженных заболеванием участках растительности;
- вычисление карты специальных областей (например, участков с развивающимся заболеванием растений) с использованием алгоритма совместной сегментации;
- получение географической привязки построенных карт и сохранение их в базе данных ГИС;
- использование полученных карт при принятии решения о необходимом количестве средств защиты растений на том либо ином участке.

В качестве данных, получаемых в реальном времени, могут использоваться:

- данные глобальной навигационной спутниковой системы;
- данные от цветной камеры видимого диапазона.

Система управления может в реальном времени корректировать данные карты особых областей, принимая более точные решения, что увеличивает эффективность решения задачи.

Сложность решения задач обработки аэрокосмических снимков обусловлена десятками гигабайт информации, передаваемых во время одного сеанса связи с искусственного спутника Земли (далее ИСЗ) или самолета. При этом требуется не только высокая суммарная производительность обрабатывающих процессоров, но и оперативная память, достаточная для одновременного хранения сотен и тысяч кадров. Даже са-

мые быстродействующие современные ЭВМ последовательной архитектуры оказываются неспособными решать задачи такой сложности.

Для решения задачи распараллеливания вычислений в системе используется понятие мультиагентных систем. Построение распределенного приложения заключается в создании графа сценариев, описывающего логическую структуру программы, а затем в его отображении на архитектуру вычислительной системы. Архитектура представляет собой множество агентов, объединенных общими средствами обмена данными. Каждый агент является равноправным участником процесса обработки и способен выполнять любую операцию из множества отдельных операций обработки, применяемых для решения задач различных предметных областей. Схема графа сценария мультиагентной системы для решения задачи параллельной обработки сельскохозяйственных аэрокосмических снимков приведена на рис. 1. Исходные данные включают совокупность кадров и параметры обработки. В результате обработки формируются карты специальных областей, на которых отмечены здоровые и пораженные участки растительности, а также оценка погрешности выполнения классификации участков растительности.



Рис. 1. Схема графа сценария системы

Распараллеливание обработки изображений

Подсистема распараллеливания обработки данных состоит из трех основных блоков.

Блок работы с изображениями содержит элементы, работающие непосредственно с файловой системой, либо ГИС, а также модули, которые позволяют читать, сохранять растровые изображения, поддерживая, если необходимо, целостность геоданных.

Блок управления выполняет функции вычисления координат для разделения (каждый растр разделяется на части, каждая из которых обрабатывается отдельным вычислительным модулем) и последующей сборки растров, а также собственно разделение и сборку.

Вычислительный кластер осуществляет параллельную обработку частей исходного изображения и содержит несколько экземпляров вычислительных модулей, осуществляющих собственно обработку дан-

ных. В их качестве выступает набор функций, предназначенных для обработки изображений, позволяющих решить поставленную задачу (например набор функций для сегментации изображения).

В качестве базы для параллельной обработки выбрана технология MPI. Эта технология выбрана из-за возможности реализации простого взаимодействия между вычислительными модулями и синхронизации [1–4].

Для передачи данных между вычислительными и управляющим модулем используется TCP-соединение, которое позволяет выполнять передачу данных по сети без потерь. Взаимодействие реализуется единообразно для каждого вычислительного модуля. Однако подобное взаимодействие может негативным образом сказываться на быстродействии системы. Чтобы снизить затраты на взаимодействие, обмен данными управляющего модуля с соответствующими вычислительными модулями реализован в виде отдельного потока выполнения [11, 13, 14].

Тестирование системы обработки изображений производилось на суперкомпьютере СКИФ К-1000. В экспериментах было задействовано от 1 до 64 вычислительных узлов, в качестве исходных данных были использованы цветные изображения размером $2\,000 \times 2\,000$ и $1\,000 \times 1\,000$ пикселей. Зависимость времени работы от количества задействованных узлов отражена на графике, приведенном на рис. 2.

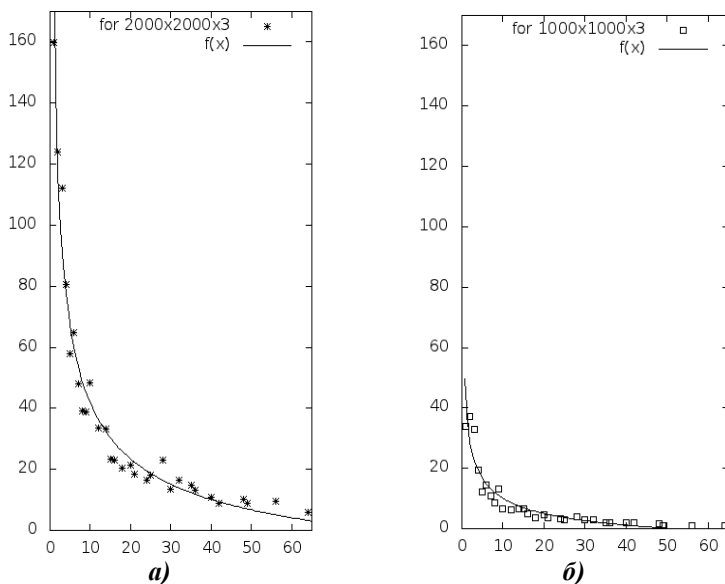


Рис. 2. График зависимости времени обработки от количества задействованных вычислительных узлов для цветных изображений размером пикселей: а) $2\,000 \times 2\,000$; б) $1\,000 \times 1\,000$

Видно, что зависимость времени работы от количества задействованных узлов пропорциональна $1/x$. Так, при обработке изображения размером $2\,000 \times 2\,000$ пикселей время работы модулей изменяется от 159 с для одного узла до 6 с на 64 узлах, а для изображения размером $1\,000 \times 1\,000$ пикселей с 34 с на одном вычислительном узле до 1 с на 64 узлах.

Заключение

Разработанная подсистема распараллеливания обработки изображений на основе интерфейса передачи сообщений MPI позволяет осуществлять прозрачное для программиста взаимодействие вычислительных узлов кластера и обмен данными между ними. Она предназначена для сокращения времени обработки данных ДЗЗ и построена на основе параллельной обработки данных на вычислительном кластере ОИПИ НАН Беларуси. Практическая значимость данной разработки заключается в возможности повышения производительности вычислений в задачах обработки данных при выполнении мониторинга.

1. Митчелл, М. Программирование для Linux. Профессиональный подход / М. Митчелл, М. Оулдем, А. Самьюэл. – М. : Издательский дом «Вильямс», 2003. – 288 с.
2. Антонов, С.А. Параллельное программирование с использованием технологии MPI : учеб. пособие / С.А. Антонов. – М. : Изд-во МГУ, 2004. – 71 с.
3. Matthew, N. Professional Linux programming / N. Matthew, R. Stones. – Berkeley : Apress, 2004. – 1058 p.
4. Hwang, K. Scalable Parallel Computing Technology, Architecture, Programming / K. Hwang, Z. Xu. – USA : McGraw-Hill, 1998. – 832 p.

Ю.В. Гурьев¹, И.В. Ткаченко², Е.И. Якушенко²

ИСПОЛЬЗОВАНИЕ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ЗАДАЧАХ КОРАБЕЛЬНОЙ ГИДРОДИНАМИКИ

¹ *Военно-Морской политехнический институт, г. Санкт-Петербург,*

² *Санкт-Петербургский государственный морской
технический университет, г. Санкт-Петербург,
igor.v.tkachenko@mail.ru*

Введение

Гидромеханика явилась одной из первых естественных наук, в которой начали широко использоваться компьютеры. Это связано с тем, что математические модели течений жидкостей и газов являются одни-

ми из наиболее сложных в физике, и аналитическое решение их уравнений за редким исключением оказывается невозможным. Применение компьютеров открыло направление фундаментальных и прикладных исследований течений – вычислительную гидродинамику (ВГД). Однако первые шаги ВГД были относительно скромными: невысокая производительность компьютеров первых поколений позволяла рассчитывать лишь невязкие потенциальные потоки и вязкие течения в приближении пограничного слоя. Появление высокопроизводительных компьютеров и технологий параллельных вычислений позволило перейти к расчету вязких потоков без привлечения принципиальных упрощений.

Однако вычислительной мощности даже самых современных суперкомпьютеров все еще недостаточно для интегрирования наиболее строгих на сегодняшний день уравнений гидродинамики – уравнений Навье-Стокса. Поэтому для расчетов натурных течений используются их осредненные аналоги – уравнения Рейнольдса (осреднение по времени) и метода крупных вихрей (осреднение по объему). Исследования, показали, что для получения адекватных данных о тонкой структуре сложных и, тем более, нестационарных течений необходимо использовать метод крупных вихрей. В настоящей работе для их замыкания использовалась модель Смагоринского с пристеночными функциями.

В докладе приводятся результаты вычислительных экспериментов по изучению движения различных морских подводных объектов (МПО) в вязкой стратифицированной среде, ограниченной свободной поверхностью, и получению их гидродинамических характеристик. Расчеты проводились с помощью коммерческого пакета Fluent и университетского пакета FlowFES [1] на двух вычислительных кластерах T-Forge32 и HP BLc3000 Twr CTO Enclosure с пиковой производительностью 256 GFlops и 300 GFlops соответственно.

Основные результаты численного моделирования

Одной из важных задач корабельной гидродинамики является определение сил и моментов, действующих на тело со стороны жидкости. На рис. 1 показаны коэффициенты нормальной силы $C_n = 2Rn/(\rho U^2 \pi R^2)$ и момента $C_m = 2Rn/(\rho U^2 \pi R^2 L)$, где Rn и M – сила и момент, ρ – плотность жидкости, U – скорость набегающего потока, R – радиус лобового сечения тела, L – его длина, в зависимости от угла атаки α для эллипсоида вращения удлинением 6:1 при числе Рейнольдса $Re = 4.2 \cdot 10^6$. На графике также представлены экспериментальные данные [2-3] и результаты расчетов, в основе которых лежат уравнения Рейнольдса, дополненные моделями турбулентности СпалартаАллмараса (S-A) и SST [4].

Относительно оценки интегральных гидродинамических характеристик можно утверждать, что как метод крупных вихрей, так и модели

на основе осреднения Рейнольдса предсказывают значения нормальной силы C_n и момента дифферента C_m практически с одинаковой степенью достоверности. На малых углах атаки модели Рейнольдса несколько занижают силу и завышают момент, в то время как метод крупных вихрей дает значения весьма близкие к экспериментальным. На больших углах атаки оба подхода занижают силу и завышают момент. На рис. 1 вертикальными линиями показаны среднеквадратичные отклонения коэффициента нормальной силы от среднего значения, полученные расчетом по методу крупных вихрей. Большие среднеквадратичные отклонения обусловлены сильным осциллирующим характером поведения коэффициента силы во времени. Следует отметить, что экспериментальные данные лежат внутри доверительного интервала метода крупных вихрей.

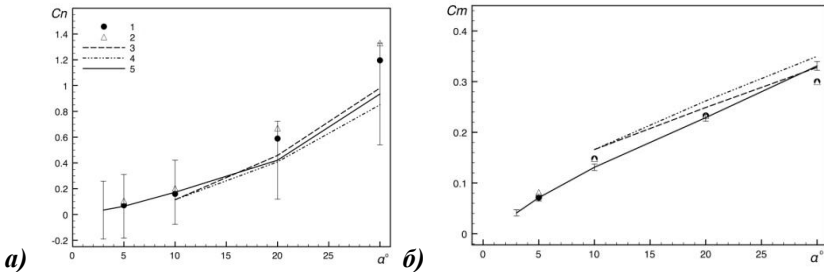


Рис. 1. Коэффициенты нормальной силы C_n (а) и момента дифферента C_m (б) в зависимости от угла атаки: 1 – эксперимент [2], 2 – эксперимент [3], 3 - SST URANS модель [4], 4 – S-A URANS модель [4], 5 – настоящий расчет

Кроме силовых, для проектирования значительный интерес представляют и распределенные гидродинамические характеристики, например поля скоростей и давлений в диске гребного винта. На рис. 2 показаны мгновенные траектории движения жидких частиц при обтекании эллипсоида под углом атаки $\alpha=20^\circ$ к набегающему потоку. Наблюдается развитая картина вихреобразования в кормовой оконечности тела. Углы схода вихрей, полученный в расчетах хорошо согласуются с экспериментальными данными.

В настоящее время при проектировании МПО используются данные о гидродинамических реакциях, полученные при движении в однородной покоящейся среде. Однако натурные условия существенно сложнее. В реальности объекты движутся в неоднородной по плотности морской воде, при воздействии поверхностных и внутренних волн, течений и ряда других гидрофизических факторов. Физическое моделирование этих условий и их влияния на гидродинамику и динамику МПО затруднительно, но ма-

тематическое, с использованием суперкомпьютеров, вполне возможно. Большой практический интерес представляет взаимодействие МПО со слоем скачка плотности морской воды – пикноклином. Внутри этого слоя возникают внутренние волны как естественного происхождения, например, создаваемые поверхностным волнением, так и искусственного, вызываемые движением кораблей и других объектов.

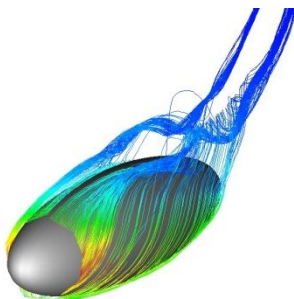


Рис. 2. Структура обтекания эллипсоида при угле атаки 20°

Для решения ряда прикладных корабельных задач актуальным является исследование гидродинамических возмущений, вносимых объектом, движущимся в стратифицированной по плотности морской среде, в частности исследование взаимодействия тела с пикноклином и природными внутренними волнами. Вычислительный эксперимент показывает, что в зависимости от относительного расположения твердого тела и пикноклина и значений критерия подобия Фруда картина образования и развития внутренних волн, а также характеристики порождаемой турбулентности могут существенно отличаться. На рис. 3 показаны результаты компьютерного моделирования взаимодействия эллипсоида с одиночной внутренней волной (солитоном). Вытесняющее воздействие тела приводит к смене знака солитона – положительная амплитуда природной волны до встречи с телом сменяется на отрицательную после его прохождения. Другие результаты исследования подобных задач приводятся в [5].

Суперкомпьютерные технологии позволили перейти к решению комплексной задачи гидродинамики и динамики корабля. Помимо определения сил и моментов, действующих со стороны жидкости на тело, в рамках модели вязкой жидкости стало возможным определение траектории движения МПО. На рис. 4 показаны результаты моделирования всплытия понтона. Полученная картина динамики тела качественно согласуется с наблюдаемой в эксперименте.

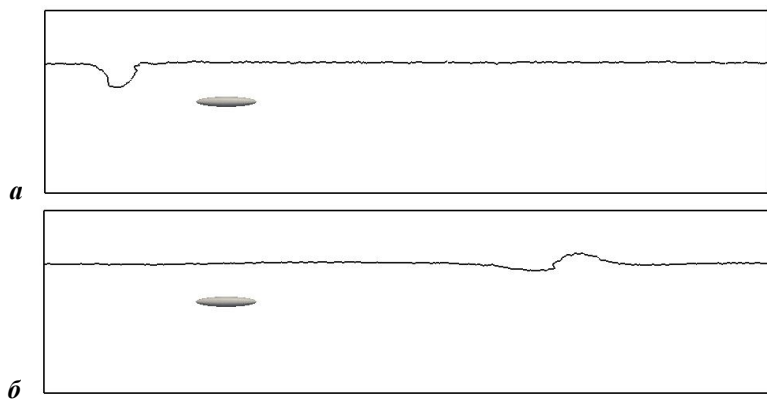


Рис. 3. Взаимодействие тела с солитоном: а – до встречи с телом, б – после прохождения тела

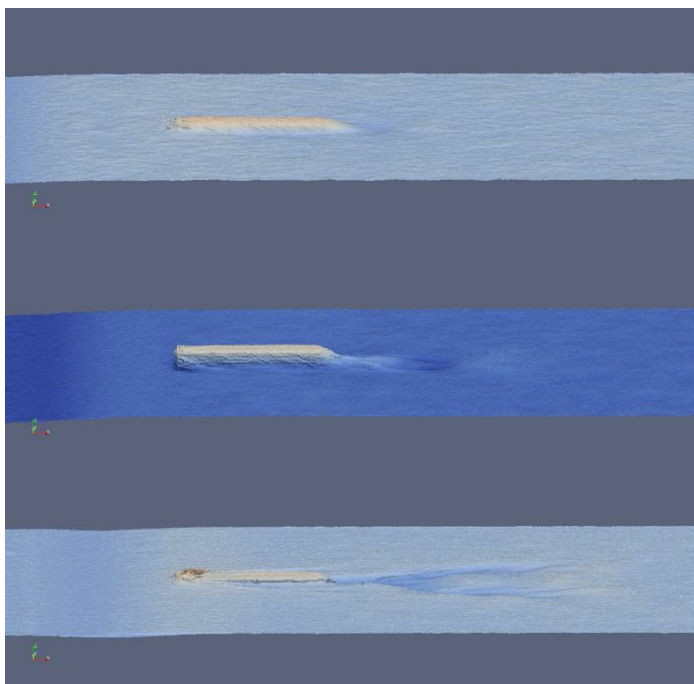


Рис. 4. Положение тела относительно свободной поверхности в различные моменты времени

Заключение

Повышение производительности компьютеров в сочетании с развитием математических моделей позволило перейти к практической реализации междисциплинарного моделирования (гидродинамика, динамика, прочность, гидрофизические поля и др.), необходимого для решения актуальных задач корабельной гидродинамики и гидрофизики.

1. Гурьев Ю.В., Ткаченко И.В. Компьютерные технологии в корабельной гидродинамике. Монография. СПб., Военно-морской инженерный институт, 2010.
2. Wetzel T., Simpson R. Unsteady three-dimensional cross-flow separation measurements on a prolate spheroid undergoing time-dependent maneuvers // Twenty-First Symp. Naval Hydrodynamics. 1997. P. 161-176.
3. Tsai C.Y., Whitney A.K. Numerical study of three-dimensional flow separation from a 6:1 ellipsoid // AIAA Paper 99-0172. 1999.
4. Kim S.E., Rhee S.H., Cokljat D. High-Incidence and Dynamic Pitch-Up Maneuvering Characteristics of a Prolate Spheroid – CFD Validation // 24th Symp. Naval Hydrodynamics. 2003. P. 609-623.
5. Гурьев Ю.В., Ткаченко И.В., Якушенко Е.И. Компьютерные технологии корабельной гидромеханики: состояние и перспективы // Фундаментальная и прикладная гидрофизика. – СПб.: Наука, 2011. – Т. 4. № 3. С. 5-13.

В.П. Дворкович, А.В. Дворкович

НОВЫЕ МЕТОДЫ ЦИФРОВОЙ ОБРАБОТКИ И СПЕКТРАЛЬНОГО АНАЛИЗА ДИНАМИЧЕСКИХ ИЗОБРАЖЕНИЙ СВЕРХВЫСОКОЙ ЧЕТКОСТИ

*ООО «Научно-производственная фирма «САД-КОМ», г. Москва,
dvp@niircom.ru*

Вслед за широким внедрением стандартного телевидения высокой четкости, имеющего относительно высокое пространственное и временное разрешение (1920 пикселей в строке; 1080 строк и 50/60 кадров/с; модель яркость/цветность 4:2:2; 16 битов информации на каждый пиксель) с объемом исходной цифровой информации порядка 2 Гбит/с, бурно развиваются перспективные системы телевидения сверхвысокой четкости и «цифрового кино», стереоскопическое и автостереоскопическое телевидение (с несколькими точками наблюдения), системы с произвольной точкой наблюдения.

Объемы исходной цифровой информации приведены в таблице.

Параметры видео высокой четкости для различных приложений

Приложения		Размер кадра, формат экрана число кадров/с	Модель цветности, кол-во битов на пиксел	Скорость циф- рового потока Гбит/с.
Телевидение сверхвысокой четкости	H0	1920x1080; 16:9; 50/60	4:2:2, 20	2,37
	H1	3840x2160; 16:9; 50/60	4:2:2, 20	9,49
	H2	5760x3240; 16:9; 50/60	4:4:4, 36	38,4
	H3	7680x4320; 16:9; 50/60	4:4:4, 36	68,3
Трехмерное ТВ: Blu-Ray: 3D-кино		2*(1920x1080); 16:9; 24	4:2:2, 32	1,58
Трехмерное ТВ: HD-3D		2*(1920x1080); 16:9; 50/60	4:2:0, 24	3,2 – 3,6
Трехмерное ТВ: MultiView 3D-TV		n *(1920x1080); 16:9; 50/60	4:2:0, 12	$n \times 1,5 \sim 100 - 1000$
Трехмерное ТВ: Free Viewpoint FTV		n *(1920x1080); 16:9; 50/60	4:4:4, 24	$n \times 1,5 \sim 100 - 1000$
Трехмерное ТВ: 3D-Integral Imaging		7680x4320; 16:9; 60	4:4:4, 36	$n^2 \times 1,5 \sim 15000$
Голографические системы ТВ		4320:8192; 60		$40 \times 1,5 \sim 10000 - 1000000$

Алгоритмы сжатия динамических изображений весьма гибки, что при различных реализациях может обеспечить существенное различие в качестве декодируемой информации при одной и той же степени ее сжатия. Стандарты ISO-IEC/MPEG, как и Рекомендации ITU-T/H Series, определяют цифровой поток закодированного сигнала, неявно задавая алгоритмы кодирования и декодирования информации. Принципы формирования цифрового потока, заложенные в стандартах MPEG, используются практически во всех используемых в настоящее время системах цифрового телевидения высокой четкости.

Эволюционное развитие существующих стандартов видеокодирования H.261, H.262, H.263 и H.264 представляет новый стандарт ITU-T H.265, обеспечивающий при высоком качестве кодирования формирование информационного потока при величине 0,08-0,1 бит на пиксел изображения.

В основу процесса обработки изображений во всех этих стандартах легли процедуры внутрикадрового кодирования с преобразованием блоков пикселей, квантованием компонентов и эффективным статисти-

ческим кодированием, а также принцип компенсации движения и обработка сигнала во времени с помощью интерполяции и экстраполяции.

Последовательность кадров динамического изображения разделяется на так называемые I-, P- и B-кадры. Изображения с внутрикадровым кодированием – I-кадры – обрабатываются без использования информации из других кадров. Они обычно используются как опорные для дальнейшего предсказания и обеспечивают точки доступа к кодированной последовательности, начиная с которых возможно декодирование. Изображения, кодируемые с предсказанием (P-кадры), обрабатываются более эффективно благодаря использованию для предсказания и компенсации движения предыдущих I- или P-кадров. Изображения, кодируемые с двунаправленным предсказанием (B-кадры), обеспечивают наибольшую степень сжатия, но требуют ссылок как на предыдущие, так и на последующие кадры для компенсации движения. B-кадры никогда не используются в качестве опорных для предсказания.

Стандарт **H.265/MPEG-H Part 2 HEVC – Высокоэффективное видеокодирование** – был разработан в ответ на растущие потребности повышения степени компрессии видео для различных приложений, таких как видеоконференцсвязь, цифровые системы хранения данных, ТВ вещание стандартной, высокой и сверхвысокой четкости, потоковая передача через Интернет и др. Он обеспечивает использование кодированного видео в разнообразном сетевом окружении, а также позволяет эффективно реализовывать обработку данных в кодирующих и декодирующих устройствах с многоядерной параллельной обработкой.

Стандарт H.265 позволяет обрабатывать видео в форме компьютерных данных и хранить его на различных устройствах, передавать и принимать его по существующим и будущим телекоммуникационным сетям, распределять по существующим и будущим каналам вещания.

В настоящее время существенно снизилась стоимость вычислительных мощностей и памяти, развиты различные средства сетевого обеспечения передачи кодированного видео, достигнуты успехи в усовершенствовании технологии видеокодирования.

Для разработки промышленного стандарта в 2010 г. группы экспертов VCEG (ITU-T Video Coding Experts Group – группа экспертов при Секторе стандартизации телекоммуникаций Международного союза электросвязи) и MPEG (ISO/IEC Moving Picture Experts Group – группа экспертов при Международной организации по стандартизации и Международной электротехнической комиссии) сформировали совместную группу по видеокодированию (JCT-VC, Joint Collaborative Team on Video Coding). Первая редакция этого стандарта была принята в рамках ITU-T в начале 2013 г. как Рекомендация H.265. В рамках ISO/IEC в на-

стоящее время он принят как окончательный проект международного стандарта ISO/IEC FDIS 23008-2 – MPEG-H Part 2 HEVC.

Новый стандарт предназначен для использования во множестве приложений, включая

- видеовещание (кабельное, спутниковое, наземное ТВ и т.д.);
- камкодеры;
- создание и распределение видеоконтента;
- цифровое кино;
- домашний кинотеатр;
- потоковая передача видео через Интернет, загрузка и проигрывание;
- медицинские изображения;
- мобильное видеовещание и видеосвязь;
- диалоговые службы реального времени (видеоконференцсвязь, видеотелефон, телеприсутствие и пр.);
- дистанционное видеонаблюдение;
- устройства хранения (оптические диски, цифровая ленточная видеозапись и пр.);
- беспроводное отображение.

Этот стандарт разрабатывался как общий в смысле пригодности для широкого диапазона приложений, скоростей выходного потока, разрешений, различного качества и разнообразных сервисов. При разработке стандарта были проанализированы различные требования типичных приложений, разработаны необходимые алгоритмические элементы, и все это было интегрировано в единый синтаксис.

Особые проблемы использования системы кодирования изображений в реальном масштабе времени возникают при обработке динамических изображений сверхвысокой четкости и систем трехмерного и многомерного телевидения, некоторые из которых приведены в таблице. И вот в этих случаях, безусловно, требуется использование суперкомпьютерных систем.

При разработке стандарта HEVC были решены следующие основные проблемы:

- повышение эффективности кодирования по сравнению с предшествующими стандартами;
- удобная интеграция в транспортные сети и системы;
- повышенная устойчивость к потерям данных;
- возможность реализации с интенсивным использованием параллельной обработки данных.

Основная цель нового стандарта – существенное повышение эффективности сжатия по отношению к существующим стандартам – до

50% снижения битового потока при сохранении визуального качества, обеспечиваемого в стандарте AVC.

Как и для всех предшествующих стандартов видеокодирования, принятых ITU-T и ISO/IEC, HEVC стандартизирует только структуру и синтаксис битового потока, а также ограничения, наложенные на битовый поток и его интерпретацию для получения декодированных изображений. Интерпретация задается с помощью определения семантического значения синтаксических элементов и процесса декодирования таким образом, чтобы любой декодер, соответствующий стандарту, давал одинаковый результат на выходе при поступлении на его вход битового потока, соответствующего ограничениям стандарта. Такое ограничение области действия стандарта дает максимальную свободу для оптимизации конкретных реализаций в соответствии с требованиями конкретных приложений (приемлемое качество кодирования, стоимость реализации, время разработки и пр.).

В докладе приведена общая структура кодера HEVC и изложены *Алгоритмы кодирования в стандарте HEVC*. Дополнительными возможностями повышения эффективности кодирования изображений сверхвысокой четкости использование дополнительных вейвлет-преобразований и неравномерное квантование различных областей при кратномасштабном анализе динамических изображений.

В случаях применения трехполосных вейвлет-фильтров обеспечивается возможность уменьшения объема исходной цифровой информации в два-три раза при практического отсутствия визуального изменения качества изображений ($\text{PSNR} \geq 40$ дБ).

В докладе приводятся варианты использования различной кратномасштабной обработки динамических изображений высокой четкости.

Еще одна проблема, рассмотренная в настоящем докладе, – это спектральный анализ изображений высокой четкости с использованием дискретного преобразования Фурье и высокоэффективных *равноволновых оконных функций Дольфа-Чебышева и Барсилона-Темеша*. Особенностью этих функций является равенство амплитуд всех боковых лепестков спектра окна, за счет чего достигается наибольшее подавление боковых лепестков при заданной ширине центрального лепестка либо минимальная ширина основного лепестка при заданном уровне боковых лепестков. Это окна разработаны для анализа дискретных сигналов, спектр которых обладает свойством периодичности.

Спектральные компоненты этих окон определяются через значения эквидистантных отсчетов преобразования Фурье, а их весовые коэффициенты обычно определяют методом обратного дискретного преобразования Фурье от спектральных характеристик.

Авторами настоящего доклада доказано, что нормированные спектральные функции окон Дольфа-Чебышева и Барсилона-Темеша, зависящие от двух параметров – n и h , характеризующих количество на интервале определения $[-1/2, 1/2]$ и величину боковых лепестков, тождественно определяются конечным числом косинусоидальных функций.

При этом, если коэффициент $n = 2m$ – четная величина, то при аппроксимации используются m четных косинусоидальных функций, а если $n = 2m-1$ – при аппроксимации используются m нечетных косинусоидальных функций.

Разработанные структуры спектральных функций окон Дольфа-Чебышева и Барсилона-Темеша позволили описать их временные характеристики, определить форму АЧХ соответствующих фильтров и их импульсные характеристики.

Д.Н. Змеев, А.В. Климов, Н.Н. Левченко, А.С. Окунев

ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНОЙ ПОТОКОВОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ «БУРАН» ДЛЯ ЗАДАЧ МОЛЕКУЛЯРНОЙ ДИНАМИКИ*

*Институт проблем проектирования в микроэлектронике РАН,
г. Москва,
zmejevdn@ippm.ru, klimov@ippm.ru, nick@ippm.ru, oku@ippm.ru*

Примером одной из областей, демонстрирующей активное использование параллельных вычислений, является молекулярное и атомистическое моделирование [1].

На суперкомпьютерах узким местом в алгоритмах задач молекулярной динамики при масштабировании является наличие глобальных барьеров, связанных с организацией пересчета списков соседей. Вследствие этого актуальным становится создание полностью асинхронного алгоритма, который базируется исключительно на локальных взаимодействиях между процессорами, обрабатывающими соседние области.

Схожие попытки предпринимались другими исследователями, при этом, правда, использовались традиционные подходы программирования взаимодействий между процессами на кластерах: MPI, специальная RDMA библиотека [2], JavaRMI [3].

В отличие от предыдущих решений в нашей работе предлагается использовать потоковую модель вычислений с динамически формируемым контекстом, которая базируется на «односторонних» сообщениях –

* Данная работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 13-07-12194.

токенах. Токен представляет собой структуру данных, содержащую, помимо самого данного, ряд служебных полей и контекст, однозначно определяющий положение данного в виртуальном адресном пространстве задачи.

Потоковая модель вычислений реализуется параллельной потоковой вычислительной системой (ППВС) «Буран» [4]. Модель вычислений основана на активации вычислительных квантов по готовности данных. Вычислительный квант представляет собой программу, которая, будучи активированной, выполняется до конца без привлечения дополнительной информации, то есть без приостановки процесса вычисления на подкачку дополнительных внешних данных. Различные вычислительные кванты между собой взаимодействуют и сохраняют состояние только через отправку сообщений, активирующих новые кванты. В узле-отправителе определяется и передаваемое значение, и адрес получателя. В этом состоит принципиальное отличие нашего подхода от традиционного, когда вычислительный процесс сам запрашивает нужные ему данные из памяти или у других процессов [5].

Распараллеливание выполнялось по принципу пространственной декомпозиции, когда каждый процессор отвечает за обработку частиц, находящихся в своей области пространства, так называемом кубоиде. При этом использовался метод «средней точки» (midpointmethod), предложенный в работе [2]. Это один из методов типа «нейтральной территории», заключающийся в том, что расчет взаимодействия производится не обязательно процессором, отвечающим за одну из частиц пары. В данном методе расчет выполняется процессором, в кубоид которого попадает середина отрезка, соединяющего пару частиц. Как показано в [2], в этом варианте объем импортируемой информации о частицах минимален. В методе «средней точки» в каждом цикле моделирования в кубоид импортируются состояния всех частиц из области импорта, которая простирается не далее чем на 26 соседних кубоидов.

Расчет сил проводится с использованием потенциала Леннарда-Джонса. Учитываются пары частиц, удаленные друг от друга на расстояние не более радиуса отсечения R_c . Предполагается, что $b < R_c < 2 \cdot b$, где b – длина стороны кубоида. Уравнения движения интегрируются методом Верле.

В ППВС «Буран» достигается уровень зернистости до частиц, при этом состояние каждой частицы передается отдельным токеном. Частица сначала передается двум соседям по X, затем при необходимости каждая копия передается соседям по Y, затем по Z. Это экономит нагрузку

на каналы передачи данных коммуникационной сети, организованной по принципу 3-мерного тора.

Силы сначала суммируются по частицам, после чего передаются, если требуется, обратно в «процессор-владелец» каждой частицы. Миграция частиц между процессорами объединена с фазой импорта частиц и не требует отдельной коммуникации.

Поскольку каждая частица передается отдельно, синхронизация передач обеспечивается счетчиками.

В «процессоре- владельце» подсчитывается по направлениям число отправляемых в каждый узел частиц, где после их прихода завершается расчет и предварительное суммирование сил. Приходящие обратно в «процессор-владелец» силы окончательно суммируются, причем завершение суммирования определяется асинхронно по разным частицам, что позволяет сразу начинать расчет уравнения движения и последующий цикл для каждой частицы независимо от других.

Текущий этап работы нацелен на определение возможностей и эффективности применения системы ППВС «Буран» для задач молекулярной динамики. Будет проведена серия экспериментов как на программной блочно-регистровой модели, так и на эмуляторе ППВС на кластерной вычислительной системе.

Имеющиеся в наличии данные позволяют говорить о том, что ППВС «Буран» является достаточно эффективной вычислительной системой для прохождения на ней задач из области молекулярной динамики.

1. Янилкин А.В., Жилиев П.А., Куксин А.Ю., Норман Г.Э., Писарев В.В., Стегайлов В.В. Применение суперкомпьютеров для молекулярно-динамического моделирования процессов в конденсированных средах // Вычислительные методы и программирование. 2010. Т. 11. С. 111-116.
2. Bowers, K.J., Chow, E., Xu, X., Dror, R.O., Eastwood, M.P., Gregersen, B.A., Klepeis, J.L., Kolossvary, I., Moraes, M.A., Sacerdoti, F.D., Salmon, J.K., Shan, Y., Shaw, D.E. Scalable Algorithms for Molecular Dynamics Simulation on Commodity Clusters. Proceedings of the ACM/IEEE Conference on Supercomputing (SC06), Article No. 84, New York, NY: IEEE, 2006.
3. Mederski, J., Mikulski, L., Bala, P. Asynchronous Parallel Molecular Dynamics Simulations. Lecture Notes in Computer Science Volume 4967, 2008, pp 439-446.
4. Стемповский А.Л., Климов А.В., Левченко Н.Н., Окунев А.С. Методы адаптации параллельной потоковой вычислительной системы под задачи отдельных классов // Информационные технологии и вычислительные системы. 2009. № 3. С. 12-21.
5. Климов А.В., Левченко Н.Н., Окунев А.С. Преимущества потоковой модели вычислений в условиях неоднородных сетей // Информационные технологии и вычислительные системы. 2012. № 2. С. 36-45.

СИНТЕЗ КОНФИГУРАЦИОННЫХ ФАЙЛОВ НА КЛАСТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог,
svetoch@pisem.net*

В настоящее время при решении задач различных предметных областей, таких как математическая физика, линейная алгебра и символьная обработка, всё больше внимания уделяется реконфигурируемым вычислительным системам (РВС), которые по сравнению с кластерными вычислительными системами (КВС) доказали свою высокую эффективность за счет адаптации задачи к архитектуре РВС. В то же время, программирование РВС является сложным, трудоемким процессом. Основное время при этом занимает синтез файлов конфигураций для программируемых логических интегральных схем (ПЛИС). Время синтеза загрузочного файла для одного современного кристалла ПЛИС, содержащего 50-60 млн. эквивалентных вентилях, может достигать до 5-7 суток. Для сокращения времени синтеза файлов конфигураций ПЛИС используются кластерные вычислительные системы. При этом сам процесс взаимодействия с кластерными системами является довольно специфическим, требующим наличия у программиста специальных знаний в этой области. Автоматизация процесса синтеза загрузочных файлов ПЛИС позволит снизить требования к квалификации программиста и сократить время получения результирующих загрузочных файлов конфигураций ПЛИС.

Для автоматизации синтеза загрузочных файлов ПЛИС необходимо использовать универсальный и в то же время эффективный способ передачи информации между компьютерами разработчиков и КВС, удовлетворяющий следующим условиям:

- соответствию стандартам стека протоколов ТСП/IP;
- возможности обмена файлами;
- возможности доступа к удалённой файловой системе;
- реализации удалённого вызова команд;
- максимально простому способу интеграции в существующие системы.

Проведенный анализ сетевых протоколов показал, что наиболее эффективным протоколом для взаимодействия с кластерными системами является протокол SSH [1], полностью удовлетворяющий всем условиям и не требующий дополнительных изменений в настройках кластерных систем.

Этапы автоматизации синтеза загрузочных файлов ПЛИС в самом общем виде:

- создание файлов конфигураций ПЛИС;
- копирование файлов конфигураций ПЛИС с машины программиста в файловое хранилище кластерной системы;
- выполнение на КВС предварительно подготовленного bash-скрипта для запуска синтеза загрузочных файлов конфигураций ПЛИС;
- копирование результирующих загрузочных файлов конфигураций ПЛИС на компьютер разработчика.

Для связи с КВС создается учетная запись с выделенным файловым пространством. Данные для авторизации указываются в коде основной программы и нигде в открытом виде не хранятся. Для соединения с КВС, выполнения операций копирования файлов и удаленного выполнения команд на КВС будет использоваться расширение PowerShell - SSH from PowerShell, содержащее библиотеку SSH.NET.

Необходимые для обработки файлы передаются с персонального компьютера пользователя на КВС, далее посредством bash-скрипта передаются параметры для обработки загруженных данных, и полученный результат загружается на персональный компьютер пользователя.

Представленный способ автоматизации процесса получения загрузочных файлов конфигураций ПЛИС позволяет снизить профессиональные требования к программисту в области КВС и сокращает время подготовки данных для синтеза загрузочных файлов ПЛИС на КВС в 2 раза.

1. Эви Немеет, Гарт Снайдер, Скотт Сибасс, Трент Р.Хейн, «UNIX Руководство системного администратора», 3-е изд., издательская группа BNV, Санкт-Петербург, 2004 г.
2. Воеводин Вл.В., Жуматий С.А. Вычислительное дело и кластерные системы. - М.: Изд-во МГУ, 2007. - 150 с.

Л.А. Крукиер, А.Л. Чикин, Л.Г. Чикина

ИСПОЛЬЗОВАНИЕ МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ ПРИ МОДЕЛИРОВАНИИ НАВОДНЕНИЯ В ДЕЛЬТЕ ДОНА

*Южный федеральный университет,
Институт аридных зон ЮНЦ РАН, г. Ростов-на-Дону,
chikin@sfedu.ru*

23-24 марта 2013 г. в результате сильного штормового нагона воды из акватории Азовского моря значительная территория в дельте Дона чрезвычайно быстро была затоплена водой. По метеорологическим наблюдениям, полученным с береговой научно-экспедиционной базы

Южного научного центра РАН на период с 20 марта по 26 марта 2013 года было промоделировано развитие нагонного явления в дельте Дона.

Постановка задачи

При моделировании использовалась двухслойная математическая модель гидродинамики водоема, достаточно адекватно описывающая ветровые течения в Азовском море в целом, и в Таганрогском заливе в частности [2]. Данные водоемы имеют как относительно глубоководные районы, так и обширные районы мелководья.

Уравнения модели решаются конечно-разностными методами. Алгоритм вычисления параметров течения воды на $(n+1)$ -ом временном слое основан на том принципе, что каждое уравнение является «определяющим» для своего неизвестного. Все остальные переменные считаются известными и берутся с n -го слоя. При конечно-разностной аппроксимации уравнений количества движения используются неявные "противопотоковые" схемы. Перепад уровня воды и вертикальный компонент скорости определяются из разностных аналогов уравнений дифференциальных уравнений.

Для решения систем линейных алгебраических уравнений, возникающих при дискретизации исходных дифференциальных уравнений, использовалась библиотека параллельных подпрограмм Aztec. В этой библиотеке реализован набор итерационных методов Крылова для решения систем линейных алгебраических уравнений с разреженными матрицами. Распараллеливание выполнено в парадигме «распределенной памяти» с использованием коммуникационной библиотеки MPI. Такой подход не только ускоряет вычисления, но и значительно снижает требования к объему оперативной памяти на вычислительных узлах. При увеличении размера сетки при дискретизации решаемой системы уравнений достаточно просто увеличить количество вычислительных узлов для решения задачи. Особенностью итерационных методов является многократное использование операции умножения матрицы на вектор. Профилирование программы показало, что на эту процедуру приходится более 50% всех вычислительных затрат [1]. Поэтому оптимизации параллельной версии этой подпрограммы для минимизации объемов пересылаемых данных было уделено большое внимание. Тем не менее, из-за большого удельного веса коммуникационных операций при работе с разреженными матрицами трудно рассчитывать на высокую степень масштабируемости этой процедуры и соответственно всей программы в целом. Кроме того, прямое тестирование подтвердило, что использование этих методов предъявляет повышенные требования к коммуникационной среде.

На задаче расчета течений было протестировано время счета на трех вычислительных системах:

- 1) на кластере IBMX с 2-ядерными процессорами Intel Xeon 5160 и коммуникационной сетью DDR Infiniband;
- 2) на кластере INFINI с процессорами Intel Pentium 4 3.4 ГГц и коммуникационной сетью SDR Infiniband;
- 3) на рабочей станции QUAD с 4-ядерным процессором Q6600 с тактовой частотой 2.4 ГГц и оперативной памятью 4Гбайт. Использовалась разностная сетка с максимальным числом неизвестных $n=1700000$. При тестировании расчет проводился для 200 временных шагов.

Как и следовало ожидать, наибольшее ускорение (Sp) прослеживается на кластере IBMX, с коммуникационной сетью DDR Infiniband. Линейное ускорение на этой платформе сохраняется для числа узлов, в то время как на кластере с однократной скоростью Infiniband ускорение начинает отклоняться от линейного уже для $n_p \leq 4$. И совершенно неоправданным представляется запуск нескольких MPI процессов на многоядерных вычислительных узлах (рабочая станция QUAD).

Численное исследование показало, что при использовании технологии MPI на вычислительной системе QUAD с 4-ядерным процессором производительность немного возрастает при подключении 2-го ядра, а затем падает при подключении последующих, в то время как на системах с распределенной памятью INFINI и IBMX наблюдается хороший рост ускорения при увеличении числа вычислительных узлов. Заметим, что на вычислительном кластере IBMX с 2-ядерным процессором на каждом узле запускался только один счетный процесс. Использование второго ядра на малом числе узлов приводило к небольшому ускорению работы программы, а при увеличении числа узлов – также, как и на системе QUAD, к замедлению работы программы.

Качество алгоритма оценивается совпадением расчетных значений с натурными данными, полученными на метеопосту в п. Донской (рис.1).

Расчет проводился с шагом по времени $dt = 5$ минут, и данные снимались через каждые 3 часа. Видно, что модель достаточно хорошо описывает развитие гидрологической ситуации в данном районе.

Для расчета объема поступающей в дельту Дона воды (куб.м/с) из Таганрогского залива было рассмотрено поперечное сечение восточной части залива длиной около 30 км. Считалось, что на затопление расходуется только нагонная составляющая перепада уровня, которая с учетом скорости течения давала объемный расход поступающей воды.

На рис. 1 **Ошибка! Источник ссылки не найден.**приведен график изменения объема поступающей из Таганрогского залива воды в дельту Дона.

Видно, что максимальный объем воды поступал в ночь с 23 на 24 марта и составил около 10 тыс. куб. м в секунду. Затем поступление стало резко уменьшаться, но все равно скорость поступления воды оставалось положительной, что вело к продолжению процесса затопления. И только к вечеру 24 марта поступление воды прекратилось, и вода стала убывать. Вечером 25 марта расход убывающей в Таганрогский залив воды составил примерно 5 тыс. куб.м/с.

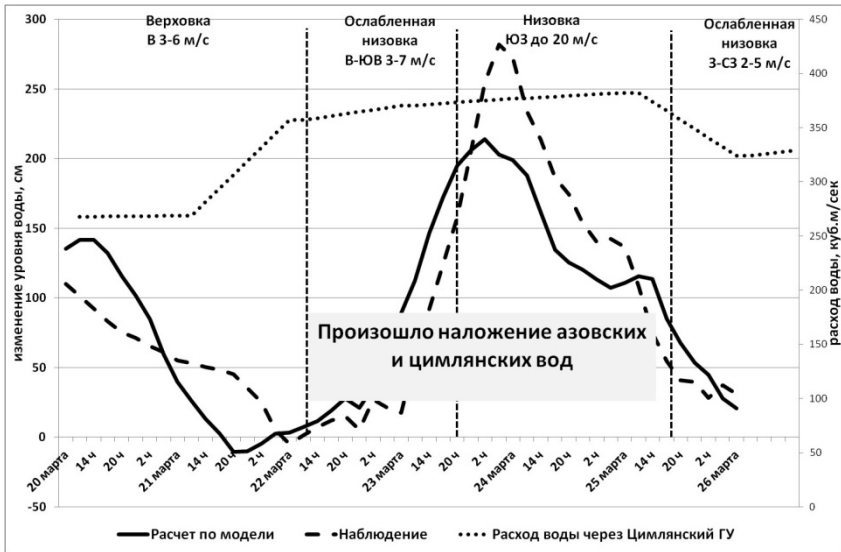


Рис. 1. Изменение относительного уровня воды на период с 20 по 26 марта 2013 года на посту «Кагальник»

С помощью моделирования установлено, что резкая смена сгонного эффекта на нагонный увеличивает скорость поступления воды, а, следовательно, и степень затопления дельты. Сравнивались две ситуации :

- а) начало нагона при юго-западном ветре после штилевой погоды и
- б) начало нагона после сгонного явления, то есть после действия восточного ветра. Сгон моделировался при скорости восточного ветра 9 м/с, который затем изменялся на ветер юго-западного направления скоростью 15 м/с.

В случае предварительного сгона значительно увеличивается скорость нагонного течения, а соответственно, и объем поступающей в дельту воды. Хотя в обоих случаях затопление дельты длилось 22 – 23 часа, что соответствует динамике повышения уровня 23-24 марта, изме-

нения уровня воды существенно различаются. Если нагон начинается при штилевой погоде, то уровень воды поднимается от нулевой отметки, соответствующей естественной глубине водоема (рис. 2), и дальше монотонно возрастает по мере поступления излишков воды. Если перед нагоном наблюдался сгон воды, то сначала уровень отрицателен (что характерно для сгонного явления), но затем быстро увеличивается и, в конце концов, становится выше нагона после штиля. Этот результат показывает, что при резкой смене восточного ветра на западный затопление происходит быстрее и оно сильнее по масштабу, чем при постоянно действующим западном ветре.

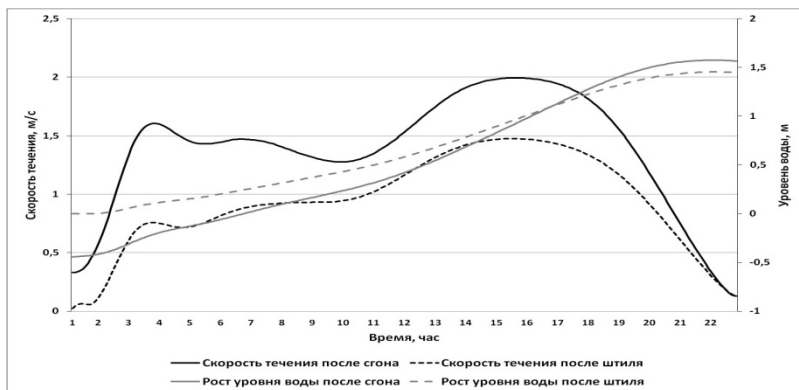


Рис. 2. Поведение скорости течения и уровня воды при различных начальных условиях нагона

Сравнение результатов расчета с наблюдаемыми данными показало, что представленная математическая модель достаточно адекватно описывает гидродинамику течений в северо-восточной части Таганрогского залива, а также сгонно-нагонные явления. Данная модель позволяет рассчитывать ветровые течения и сгоны-нагоны в случае изменения береговой линии, что позволит в дальнейшем прогнозировать степень опасности затопления дельты Дона.

1. Дацик В.Н., Крукиер Л.А., Чикин А.Л.. Реализация на высокопроизводительных вычислительных системах математической модели ветровых течений в Керченском проливе. //Вестник УГАТУ, Т.15, № 5 (45), 2011. – С.155-160.
2. Chikin A. L. A technique for evaluating flow parameters in water bodies with a highly heterogeneous depth. Water Resources, 2005, Volume 32, Number 1, January. – pp. 50-55.

МОДИФИКАЦИЯ АЛГОРИТМА ИЗВЛЕЧЕНИЯ ТЕКСТУРНЫХ ХАРАКТЕРИСТИК С ПОМОЩЬЮ ВЫЧИСЛЕНИЙ НА ГРАФИЧЕСКИХ ПРОЦЕССОРАХ*

*Томский политехнический университет, г. Томск,
wedun@tpu.ru*

Цель работы

Аэрокосмические снимки не позволяют производить анализ данных по спектральным признакам непосредственно, без использования сторонних средств. Это обусловлено сложностью структуры аэрокосмических наблюдений. Спектральные портреты объектов земной поверхности зависят от многих факторов, таких как рельеф, тип почв, климат, географическое положение местности, поэтому они не стационарны. Для повышения достоверности принимаемых решений, необходимо использовать априорную информацию о геометрии съемки и контекстную информацию изображений [1].

Предпочтительность текстурных признаков при классификации облачности заключается в потенциальных возможностях агрегировать контекстную информацию с определенными свойствами инвариантности под конкретную задачу распознавания образов.

Целью работы является повышение эффективности обработки спутниковых снимков путем разработки ядра на CUDA, позволяющего быстро получить значения текстурных характеристик с помощью фильтров Лавса.

Текстурные признаки

Для вычисления энергетических характеристик спутникового снимка используются следующие фильтры:

$$\begin{array}{ll}
 L5 \quad (\text{Level}) &= \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} \\
 E5 \quad (\text{Edge}) &= \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \end{bmatrix} \\
 S5 \quad (\text{Spot}) &= \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \end{bmatrix} \\
 W5 \quad (\text{Wave}) &= \begin{bmatrix} -1 & 2 & 0 & -2 & 1 \end{bmatrix} \\
 R5 \quad (\text{Ripple}) &= \begin{bmatrix} 1 & 4 & 6 & -4 & 1 \end{bmatrix}
 \end{array}$$

Названия векторов описывают их назначение. Вектор L5 предназначен для вычисления симметричного взвешенного локального среднего значения. Вектор E5 предназначен для обнаружения краев, S5 – для обнаружения пятен, W5 – для обнаружения волн, а R5 – для обнаружения образа в виде ряби.

* Работа поддержана грантами Российского фонда фундаментальных исследований №14-07-31090 и №14-0731018.

Перечисленные выше вектора применяются для создания двухмерных фильтров путём комбинации двух одномерных фильтров. Двумерные маски получаются в результате умножения пар векторов. Например, для получения маски E5L5 надо умножить вектор E5 на L5:

$$\begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}.$$

Полученный фильтр E5L5 позволяет детектировать изменение границ по вертикали и изменение уровня по горизонтали.

На первом шаге в методе Лавса устраняется влияние интенсивности освещения. Для этого по изображению перемещается небольшое окно и из значения каждого пикселя вычитается локальное среднее значение. В результате данной предварительной обработки формируется изображение, на котором средняя интенсивность каждой пиксельной окрестности близка к 0. Выбор размера окна зависит от типа анализируемого изображения.

После предварительной обработки к полученному изображению применяется каждая из масок Лавса [2].

Алгоритмы фильтрации

Одной из проблем обработки графических изображений является выполнение преобразований на границах снимка. Существует несколько подходов, каким образом произвести фильтрацию по краю: инициализация пикселей, попадающих в фильтрующее окно, но расположенных за пределами изображения случайными значениями, средними, зеркальным отражением граничных пикселей, с дополнительным ослаблением интенсивности пикселей для систем компьютерного зрения. В нашей работе был взят подход зеркального отражения [3].

Матрица, полученная из исходной матрицы интенсивности добавлением дополнительных зеркальных FSize строк и FSize столбцов, будет называться расширенной. Таким образом, для обработки краевых регионов необходимо создать расширенную матрицу и получить в результате фильтрации матрицу, размером совпадающую с исходной. Однако выделение памяти под дополнительную матрицу будет также недостатком подхода, т.к. она в большей степени копирует значения исходной матрицы интенсивности и при больших размерах снимка сильно загружает оперативную память. Другим вариантом обработки при вызове ядра CUDA может быть анализ расположения фильтрующего региона: если область не краевая, то происходит обычная обработка, в противном случае добавляются фиктивные пиксели в зависимости от удаленности от границы изображения.

Для вычислений необходимо ввести две переменные PH и PW , позволяющие определить, какие значения из исходной матрицы будут браться для вычисления значений фиктивных пикселей:

$$PH = 2 \times H + 0.5 \times FSize,$$

$$PW = 2 \times W + 0.5 \times FSize,$$

где W и H – ширина и высота исходного изображения, $FSize$ – размер фильтра.

Фильтруемый регион однозначно определяется координатами верхнего левого угла i, j . Координаты пикселей ($Xcoordinate$, $Ycoordinate$) используемых при операции свертки внутри одного окна:

$$Xcoordinate = j + fj - 0.5 \times FSize, fj = 0..FZise - 1,$$

$$Ycoordinate = i + fi - 0.5 \times FSize, fi = 0..FZise - 1.$$

Отрицательные значения $Xcoordinate$ и $Ycoordinate$ означают, что регион является граничным и обрабатываемый пиксель – фиктивный, расположенный за пределами изображения, т.е. для него необходимо получить значение из исходной матрицы. Значение для такого фиктивного пикселя берется из элемента входной матрицы, расположенного на новой позиции ($Xcoordinate$, $Ycoordinate$):

$$Xcoordinate = 0.5 \times FSize - j - fj - 1,$$

$$Ycoordinate = 0.5 \times FSize - i - fi - 1.$$

Может получиться, что значения координат выходят за границы изображения при $Xcoordinate > W$ или $Ycoordinate > H$. В этом случае опять имеет место фиктивный пиксель и координаты элемента исходной матрицы, соответствующие ему, находятся с помощью объявленных ранее величин PH и PW :

$$Xcoordinate = PW - j - fj - 1,$$

$$Ycoordinate = PH - i - fi - 1.$$

Таким образом, алгоритм вычисления текстурных характеристик Лавса на CUDA можно записать следующими шагами (Вариант А).

Шаг 1. Копирование исходной матрицы в память GPU.

Шаг 2. Вызов ядра фильтрации матрицы для вычисления текстурных характеристик Лавса, в т.ч. вычисляющего значения фиктивных пикселей.

Обработка граничных регионов предполагает использование условных операторов, обрабатывающих фиктивные пиксели, что отсутствует в случае с использованием расширенной матрицы. Наличие условий снижает в определенной степени производительность потоков GPU. Поэтому необходимо также оценить производительность без конвертации координат, но с дополнительной памятью (Вариант В) – тот путь, который был предложен ранее.

Вариант В, предполагающий использование расширенной матрицы, можно описать следующими шагами.

Шаг 1 Выделение памяти под расширенную матрицу.

Шаг 2. Копирование расширенной матрицы в память GPU.

Шаг 3. Вызов ядра, выполняющего фильтрацию Лавса.

Для оценки производительности двух подходов были написаны два ядра CUDA, реализующих оба варианта А и В, позволяющих находить значения текстурных характеристик для фильтров E5L5 и W5S5. Результаты производительности обоих вариантов даны в таблице. Разработанное приложение тестировалось на GPU Nvidia GeForce GTX 460.

***Скорость обработки фрагментов снимков с помощью использования
А) преобразования координат в ядре без выделения дополнительной
памяти, В) с выделением дополнительной памяти
без преобразования координат в ядре***

Размер фрагмента снимка	400x500		1200x1600		3000x13000	
Вариант	А	В	А	В	А	В
Загрузка файла, мс	16		131		1354	
Выделение памяти, мс	94	145	109	164	172	780
Работа яд- ра, мс	1,04	0,69	9,9	6,6	200	133
Выгрузка в файл, мс	18		78		1627	
Всего, мс	129,04	179,69	327,9	379,6	3353	3894

Как видно из результатов теста, включение анализа граничных пикселей увеличивает время обработки ядра приблизительно на 50%, однако существенно снижает время, связанное с выделением памяти (т.к. не выделяется расширенная матрица). Учитывая разные порядки времени работы ядра и времени выделения памяти с заполнением дополнительной матрицы, следует заключить, что Вариант А с работой с фиктивными пикселями более предпочтителен.

1. Протасов К.Т. Непараметрический алгоритм распознавания объектов подстилающей поверхности Земли по данным аэрокосмической съемки / К.Т. Протасов, А.И. Рюмкин // Вестник Томского государственного университета. – 2002. – № 275. – С. 41–46.
2. Андреев Г.А. Анализ и синтез случайных пространственных текстур / Г.А. Андреев, О.В. Базарский, А.С. Глауберман, А.И. Колесников, Ю.В. Коржик, Я.Л. Хлявич // Зарубежная радиоэлектроника. – 1984. – №2. – С. 3–33.
3. Харалик Р.М. Статистический и структурный подходы к описанию текстур // ТИИЭР. – 1979. – Т. 67. – № 5.

**ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ
В РАДИОЛОКАЦИОННОМ КОМПЛЕКСЕ,
ИСПОЛЬЗУЮЩЕМ ЗОНДИРУЮЩИЕ СИГНАЛЫ
С ЛИНЕЙНОЙ ЧАСТОТНОЙ МОДУЛЯЦИЕЙ
И ИЗМЕНЯЮЩИМСЯ ЗНАКОМ ДЕВИАЦИИ ЧАСТОТЫ**

*Научно-конструкторское бюро цифровой обработки сигналов
Южного федерального университета, г. Таганрог,
marko@sfedu.ru; dsp@sfedu.ru*

Одной из важнейших задач, стоящих перед современными радиолокационными комплексами (РЛК) в режиме работы по скоростным целям, является проблема селекции движущихся целей (СДЦ). Без применения последних достижений суперкомпьютерных технологий в радиолокации решить указанную задачу невозможно.

В докладе дан анализ результатов разработки и исследования перспективных алгоритмов цифровой обработки сигналов (ЦОС) РЛК в режиме СДЦ.

Для достижения поставленной цели необходимо:

- обосновать применение разработанных алгоритмов ЦОС РЛК в режиме работы СДЦ с использованием зондирующих сигналов с внутриимпульсной линейной частотной модуляцией (ЛЧМ), чередованием знака девиации от импульса к импульсу и применением череспериодной компенсации огибающих (ЧПКО) выходных сигналов согласованных фильтров (СФ);

- проанализировать результаты экспериментальных исследований предлагаемых алгоритмов, заключающиеся в определении зависимостей амплитуды и протяженности выходных сигналов СФ от скоростей целей, в оценке влияния весовой обработки (ВО) на выходные сигналы, в получении зависимостей интервалов времени между максимумами на выходе ЧПКО от параметров движения объектов и в вычислении скоростных характеристик системы СДЦ с ЧПКО.

Известно, что для определения однозначной дальности в процессе поиска целей целесообразно применять импульсные зондирующие сигналы с низкой частотой повторения (НЧП) импульсов [1]. При этом должны обеспечиваться не только однозначная дальность, но и заданная для РЛК дальность обнаружения. Этим требованиям удовлетворяют сложные импульсные сигналы с большой базой, в частности сигналы с внутриимпульсной ЛЧМ, девиация частоты которых позволяет получить требуемое разрешение. Период повторения зондирующего импульса должен превышать заданный диапазон измерения времени запазды-

вания эхосигналов относительно момента начала излучения. Допустимая длительность ЛЧМ импульсов ограничивается требованиями к минимальной дальности обнаружения.

Выбор для обзора воздушной обстановки зондирующих сигналов с НЧП обусловлен тем, что при поиске воздушных целей для оценки тактической обстановки в секторе обзора РЛК важно прежде всего получить первичную информацию о дальности. Также желательно получить информацию о скорости и направлении движения цели.

Помехозащищённость любой современной РЛС должна быть обеспечена как при пассивных, так и при активных помехах различного происхождения [2].

Эффективным методом защиты от непрерывных и прицельных по частоте шумовых помех является перестройка несущей частоты, заставляющая помехопостановщика перейти от прицельной по частоте шумовой помехи к заградительной шумовой помехе, результативность которой существенно ниже.

Возможна перестройка несущей частоты от пачки к пачке и от импульса к импульсу. Под пачкой импульсов понимают последовательность эхосигналов, полученных от цели за время контакта с целью в каждом угловом положении антенной системы.

Перестройка несущей частоты от пачки к пачке применяется в импульсно-доплеровских РЛК со средней (или высокой) частотой повторения импульсов, не разрушающих когерентность обработки сигналов в течение всего времени контакта с целью. Однако при воздействии прицельной шумовой помехи с малым временем реакции помехопостановщика на изменение несущей частоты подавляемой РЛК перестройка несущей частоты от пачки к пачке неэффективна, так как за это время помехопостановщик с большой вероятностью может подстроиться на рабочую частоту РЛК.

Для обеспечения помехозащиты от воздействия мешающих отражений и пассивных помех требуется применять СДЦ, которые основаны на эффекте выделения межпериодного доплеровского набега фаз, что требует обеспечения соответствующей межпериодной когерентности применяемой обработки. То есть применение перестройки несущей частоты от импульса к импульсу в режиме НЧП не позволяет использовать традиционные межпериодные системы СДЦ, основанные на определении межпериодного доплеровского набега фаз. Кроме того, такие СДЦ имеют существенный недостаток, так называемый эффект «слепых скоростей» [1, 2].

Чтобы повысить эффективность применения перестройки несущей частоты, её необходимо осуществлять в РЛК от импульса к импульсу. В

режиме обзора это возможно при использовании сигналов с НЧП и некогерентной межпериодной обработки пачки эхосигналов [3, 4].

В предлагаемых алгоритмах ЦОС для выделения сигналов движущихся целей на фоне отражений от местных предметов или пассивных помех при использовании перестройки несущей частоты от импульса к импульсу применяются импульсные ЛЧМ сигналы, в которых меняется знак девиации внутриимпульсной частотной модуляции от импульса к импульсу. Тогда в четных и нечетных периодах повторения эхосигналы от движущихся целей будут смещены относительно истинного значения дальности в положительную или в отрицательную стороны. Величина этого временного сдвига прямо пропорциональна доплеровскому сдвигу частоты ЛЧМ сигнала, а коэффициент пропорциональности равен отношению длительности зондирующего сигнала к его девиации. Сигналы, отражённые от неподвижных и малоподвижных объектов, практически не будут отличаться по задержке и могут быть скомпенсированы путем взаимного попарного вычитания продетектированных сжатых сигналов. Результаты попарных вычитаний для каждого элемента дальности накапливаются внутри пачки некогерентно. Возможность выделения сигналов движущихся целей при попарном межпериодном вычитании обусловлена смещением сжатых сигналов по времени запаздывания (дальности) в противоположные стороны относительно истинного значения. По интервалу времени между моментами обнаружения двух эхосигналов от каждой цели вычисляются уточненная дальность и ее скорость, а по знакам эхосигналов – направление движения.

Экспериментальные исследования программных моделей алгоритмов ЦОС в системах СДЦ с ЧПКО осуществлялись с повышенной частотой дискретизации входного непрерывного сигнала для более точного анализа погрешностей методов и алгоритмов. Оценки эффективности алгоритмов РЛК в режиме СДЦ с ЧПКО делались на основе обработки типового ЛЧМ сигнала X-диапазона частот длительностью $\tau_n = 32,8$ мкс и с девиацией частоты $\Delta f_d = 2,7$ МГц. Размер графа БПФ, используемого для реализации СФ методом «быстрой» свертки, $N_{\text{БПФ}} = 4096$ [5].

Моделировались два оптимальных фильтра: СФ1, согласованный с ЛЧМ радиосигналом с возрастающей частотой (положительный знак девиации) и СФ2 – с убывающей частотой (отрицательный знак девиации). Оба фильтра согласованы с дискретными радиосигналами, полученными из непрерывного сигнала с повышенной частотой дискретизации.

Исследовалась зависимость от скорости формы огибающих откликов СФ на эхосигналы, отраженные от движущихся целей со скоростя-

ми в диапазоне от минус 1000 м/с до 1000 м/с при строгом учете эффекта Доплера.

При моделировании для огибающей сигнала каждого СФ определялись максимальное значение и его задержка относительно момента начала излучения, ширина главного лепестка сжатого сигнала по уровню 0,637 и по уровню 0, а также уровень боковых лепестков.

На этапе исследования алгоритмов ЦОС радиолокационного комплекса в режиме СДЦ экспериментально оценивалась эффективность подавления боковых лепестков сжатого сигнала путем весовой обработки (ВО) спектров ЛЧМ сигналов окном Хэмминга.

Одновременно для огибающей сжатого с применением ВО сигнала определялись уровень и задержка максимума, ширина главного и уровень боковых лепестков.

Следующей задачей исследования алгоритмов ЦОС являлось определение зависимостей характеристик СДЦ с ЧПКО от скорости движения целей.

Были исследованы зависимости интервала времени между двумя максимумами сигнала на выходе СДЦ с ЧПКО, а также их максимальные значения от скорости цели. Эксперименты проводились на программных моделях СФ с ВО и без нее.

Скорость цели изменялась в пределах от минус 1000 до 1000 м/с с шагом 25 м/с, и в каждом случае определялась задержка между равными максимумами выходного сигнала и их величина.

Результаты проведенных с повышенной частотой дискретизации экспериментальных исследований разработанных алгоритмов ЦОС показали следующие потенциальные возможности СДЦ с ЧПКО:

- величина максимума сжатого сигнала, ширина главного лепестка и уровень боковых лепестков практически (единицы процентов) не зависят от скорости при выбранных параметрах сигналов и заданных диапазонах изменения скорости целей;

- ВО с применением функции Хемминга в частотной области существенно снижает уровень боковых лепестков огибающей отклика СФ до минус 36 дБ. Однако при этом снижается уровень максимума выходного сигнала СФ до $0,56A_{\text{max}}$, а ширина главного лепестка по уровню 0,637 увеличивается в 1,77 раз;

- отклики СФ 1 и СФ 2 на сигналы с противоположным законом модуляции по форме повторяют спектры ЛЧМ сигналов, а их уровень достаточно мал по сравнению с максимумами сжатых сигналов. Отношение максимального значения сигнала, согласованного с СФ, к среднему уровню сигнала с противоположным законом модуляции на выходе того же фильтра практически не зависит от скорости цели и равно 22,5 дБ. Это является дополнительным преимуществом применения че-

редования знака девиации внутриимпульсной ЛЧМ, так как появляется возможность существенного ослабления мешающих отражений «со второй дальности»;

- зависимость интервала между максимумами на выходе СДЦ с ЧПКО выходных сигналов СФ от скорости имеет существенную нелинейность в области малых скоростей, обусловленную взаимным влиянием откликов огибающих при наложении и малым уровнем сигналов после вычитания. При использовании сигналов без ВО появляются пульсации функции, снижающие точность определения скорости цели, а зона существенного отклонения от линейной аппроксимации характеристики составляет от минус 200 м/с до 200 м/с. При использовании ВО пульсации отсутствуют, но увеличивается нерабочая зона скоростей от минус 400 м/с до 400 м/с;

- экспериментально исследованы скоростные характеристики СДЦ с ЧПКО. Без ВО характеристики имеют более крутой спад и значения сигнала в заданном диапазоне скоростей целей существенно больше, чем в ЧПКО с ВО, что позволяет более успешно решать задачи обнаружения одиночных целей.

Таким образом, разработанные перспективные алгоритмы ЦОС для РЛС с СДЦ не имеют эффекта «слепых скоростей» и позволяют уменьшить время обнаружения целей в режиме работы с НЧП до одного такта, а первичная информация о дальности, скорости и направлении движения обнаруженных целей позволит уменьшить время захвата и сопровождения целей в следующих режимах.

1. Дудник П. И., Кондратенков Г.С., Татарский Б.Г., Ильчук А.Р., Герасимов А.А. Авиационные радиолокационные комплексы и системы: учебник для слушателей и курсантов ВУЗов ВВС /Под ред. П.И. Дудника. - М.: Изд-во ВВИА им. проф. Н.Е. Жуковского, 2006. – 1112 с.
2. Бруханский А. В. Системы селекции движущихся целей: учебное пособие. – М.: МАИ, 1990.
3. Патент US 5530448 А. МКИ G01S13/528, G01S13/28, G01S13/10. Three-pulse MTI without blind speeds. Оpubл. 25.06.1996.
4. Патент US 5870054 А. МКИ G01S13/524, G01S13/28. Moving target indicator with no blind speeds. Оpubл. 09.02.1999.
5. Маркович И.И. Цифровая обработка сигналов в системах и устройствах: монография / И.И. Маркович. – Ростов н/Д: Изд-во ЮФУ: 2012. – 236 с.

ПРИМЕНЕНИЕ СУПЕРКОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ РАЗРАБОТКИ ОАО «НИЦЭВТ» В ПРИКЛАДНЫХ СИСТЕМАХ

*ОАО «Научно-исследовательский центр электронной
вычислительной техники»», г. Москва,
{director,izgalin,simonov}@nicevt.ru*

Эффективное использование суперкомпьютерных систем является неотъемлемой составляющей процесса создания высокотехнологичной и конкурентоспособной продукции в различных отраслях экономики.

Наиболее важными аспектами при разработке суперкомпьютерных систем являются создание высокопроизводительного аппаратного обеспечения (процессоры, высокоскоростная память, коммуникационная сеть, подсистемы хранения данных, инфраструктура энергообеспечения и охлаждения) и разработка методов параллельного решения прикладных задач.

В настоящее время в ОАО «НИЦЭВТ» создан значительный задел по разработке наиболее важных аппаратных компонентов и программных средств суперкомпьютеров высшего диапазона производительности [1]. Компетенции ОАО «НИЦЭВТ» в области разработки отечественных суперкомпьютерных технологий (СКТ) распространяются от разработки СБИС и вычислительных платформ до создания высокопроизводительных вычислительных систем и их программного обеспечения.

Разработанные на предприятии высокопроизводительная вычислительная платформа ЕС1740.0001 [2] и высокоскоростной интерконнект на базе СБИС ЕС8430 [3] являются ключевыми элементами для создания суперкомпьютеров и серверов для различных применений. Ведутся работы по инновационному проекту создания прикладного вычислительного комплекса с непосредственным жидкостным охлаждением (см. рисунок), обладающего целым рядом преимуществ (более равномерное распределение температуры по поверхности системных плат, повышение надежности, увеличение энергоэффективности и др.).

В дальнейшем ОАО «НИЦЭВТ» ориентируется на решение практических задач применения суперкомпьютерных систем в стационарных и бортовых комплексах для различного круга заказчиков, в том числе в интересах обеспечения безопасности государства, с учетом требований заказчиков по постоянному повышению функциональных возможностей, интеллектуализации разрабатываемой техники, снижению влияния человеческого фактора на решение задач.



***Вычислительный комплекс с иммерсионной системой
охлаждения на основе высокопроизводительной
вычислительной платформы EC1740.0001***

1. Михеев В.А., Симонов А.С. История и развитие суперкомпьютерных технологий в ОАО «НИЦЭВТ» // «Научные технологии», №1, 2014. С. 6-11.
2. Слуцкий А.И., Симонов А.С., Казаков Д.В. Универсальная высокопроизводительная вычислительная платформа «Ангара» // «Научные технологии», №1, 2014. С. 17-20.
3. Жабин И.А., Макагон Д.В., Поляков Д.А., Симонов А.С., Сыромятников Е.Л., Щербак А.Н. Первое поколение высокоскоростной коммуникационной сети «Ангара» // «Научные технологии», №1, 2014. С. 21-27.

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА РАЗНОВРЕМЕННЫХ КОСМИЧЕСКИХ СНИМКОВ

*Объединенный институт проблем информатики НАН Беларуси,
г. Минск,
turnic@newman.bas-net.by*

Введение

Современные спутниковые технологии позволяют обнаруживать чрезвычайные ситуации на местности по данным дистанционного зондирования Земли (ДЗЗ). При этом процесс обнаружения чрезвычайных ситуаций на местности по космическим снимкам включает:

- 1) доведение качества текущего снимка до уровня эталона. Под термином «текущий снимок» понимается снимок, полученный в данный момент времени;
- 2) совмещение эталонного и текущего снимков с точностью до пикселя;
- 3) обнаружение изменений на текущем снимке;
- 4) классификация выявленных изменений;
- 5) принятие решения об обнаружении чрезвычайной ситуации на местности или объекте наблюдения.

По нормативам, существующим в Белоруссии, время обнаружения и классификации чрезвычайной ситуации по данным дистанционного наблюдения не должно превышать 30 минут с момента поступления текущего снимка.

Автоматический поиск изменений на последовательности космических снимков в реальном времени требует применения высокопроизводительных вычислительных систем. Это обусловлено следующими факторами:

- средний объем панхроматического космического снимка, имеющего пространственное разрешение 1 м на пиксель, составляет 400 Мбайт;
- количество объектов мониторинга на космическом снимке может достигать нескольких десятков;
- эталонный и текущий снимки, как правило, получены в разные сезоны года и в разных условиях освещения подстилающей поверхности;
- пространственная ориентация текущего и эталонного изображения не совпадает.

Известные программные комплексы, используемые для обработки данных дистанционного зондирования Земли (ERDAS, ENVI и другие) не обеспечивают в автоматическом режиме обнаружение на местности чрезвычайных ситуаций.

Параллельная обработка данных ДЗЗ при автоматическом совмещении спутниковых изображений

Сущность технологии совмещения эталонного и текущего изображений заключается в следующем [1]. На эталонном снимке выбирается сеть реперных точек, географические координаты которых известны. Реперные точки должны быть узнаваемы в любые сезоны года, а их количество на космическом снимке может достигать трех десятков. Оператор находит и идентифицирует реперные точки на текущем снимке. Далее, устанавливается связь между системами координат текущего и эталонного снимков. После этого выполняются геометрические преобразования текущего снимка, в результате которых текущий снимок совмещается с эталонным. При этом оператор должен визуально идентифицировать опорную точку с точностью до пикселя. Опыт показывает, что визуальный поиск и ручная расстановка реперных точек являются трудоемкими и приводят к ошибкам, вызванным человеческим фактором.

Разработана технология автоматического поиска и расстановки реперных точек на космическом снимке во времени, близком к реальному, сущность которой заключается в следующем. На космическом снимке осуществляется поиск опорных участков, каждый из которых содержит реперную точку. Опорный участок должен быть уникальным для всего изображения или большей его части. В качестве критерия уникальности может выступать коэффициент корреляции Пирсона. В общем случае могут быть использованы несколько критериев уникальности опорного участка. Необходимо отметить, что критерии уникальности должны быть инвариантны к условиям освещенности подстилающей поверхности в заданных пределах угла поворота и масштаба опорного участка. От размеров опорного участка зависит время автоматического поиска и расстановки реперных точек. По этой причине предложено распараллелить процесс поиска опорных участков на эталонном снимке. Эталонный снимок разбивается на секторы прямоугольной формы, количество которых соответствует числу планируемых реперных точек. При совмещении двух изображений не обязательно знать географические координаты реперных точек. В качестве реперной точки можно принять пиксель, соответствующий геометрическому центру опорного участка. Далее, в каждом секторе эталонного снимка необходимо найти один уникальный опорный участок.

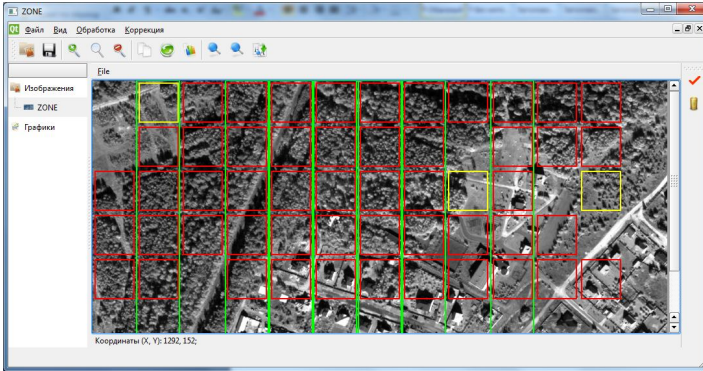
Поиск опорных участков на эталонном снимке

Пусть имеется вычислительная система, содержащая N_p процессоров. Размеры опорного участка составляют m_x и m_y пикселей по осям системы координат снимка соответственно. В качестве критерия уникальности опорного участка принимаем коэффициент корреляции Пирсона k_p . Расстояние между соседними реперными точками должно

быть не менее d_u . Необходимо в автоматическом режиме найти на эталонном снимке уникальные опорные участки в количестве N_u .

Поиск опорных участков выполняется в следующей последовательности.

1. Эталонный снимок разбивается на N_u прямоугольных секторов $Su(x,y)$. Расстояние между центрами соседних секторов не должно превышать d_u .
2. Каждый сектор разбивается на множество прямоугольных участков, размеры каждого из которых составляют $m_x \times m_y$ пикселей. В результате формируется матрица $Fu(x,y)$, содержащая $n_x \times n_y$ прямоугольных участков – кандидатов в опорные участки. На рисунке показана матрица $Fu(x,y)$.



Матрица потенциальных опорных участков сектора снимка

3. Производится параллельный поиск потенциальных опорных участков (ПОУ) в первых N_p строках матрицы $Fu(x,y)$. При этом в строке не может быть более одного ПОУ. Элемент (участок) строки считается ПОУ, если коэффициент корреляции его с остальными участками минимальный и не превосходит заданный предел. Каждому ПОУ присваивается ранг информативности $Rinf(i,j)=1-\gamma(i,j)$, где $\gamma(i,j)$ – коэффициент корреляции. Здесь (i,j) – координаты ПОУ, $i \in \overline{1, m_x}$, $j \in \overline{1, m_y}$. Далее осуществляется поиск ПОУ в последующих N_p строках матрицы $Fu(x,y)$.
4. Среди найденных ПОУ производится последовательный поиск опорного участка $Fu(k)$, где k – номер сектора. При этом корреляция участка $Fu(k)$ с остальными ПОУ должна быть минимальной.

5. Параллельно осуществляется поиск опорных участков в других секторах эталонного снимка.

Совмещение разновременных космических снимков

Текущий снимок может быть повернут относительно эталона. Максимальный угол поворота достигает 30^0 . В результате предварительной обработки космического снимка угол поворота его может быть доведен до 3^0 . При поиске опорных участков на текущем изображении необходимо учитывать угол поворота, который заранее не известен. Расстановка реперных точек на текущем снимке выполняется в следующей последовательности:

- 1) текущий снимок разбивается на N_p прямоугольных секторов $S^*u(x,y)$;
- 2) производится параллельный поиск опорных участков в выделенных секторах. В каждом секторе может находиться несколько опорных участков. При этом необходимо учитывать кроме поворота и масштаб эталонного снимка относительно текущего;
- 3) вычисляются координаты реперных точек в системе координат текущего снимка;
- 4) устанавливается связь между координатами соответствующих реперных точек на эталонном и текущем изображениях;
- 5) по известным формулам [2] вычисляются угол поворота, сдвиги и изменение масштаба текущего снимка относительно эталонного.

Алгоритмы обнаружения изменений на космических снимках приведены в [3].

1. Белозерский, Л.А., Н.И. Мурашко. Проблемы построения автоматизированных программно-информационных комплексов спутникового мониторинга локальных объектов заданных территорий // Материалы Четвертого Белорусского космического конгресса. 27–29 октября 2009 г., Минск : ОИПИ НАН Беларуси, 2009. – Т. 2. – С. 151–155.
2. Баранов В.Н., Бойко Е.Г., Краснорылов И.И. и др. – Космическая геодезия: учебник для вузов. М., Недра, 1986.
3. Белозерский Л.А., Н.И. Мурашко, Л.В. Орешкина, А.И. Шевченко. Разностные представления изменений внешнего облика наземных объектов разновременной космической съемки оптического диапазона / Донецк: Наука и образование, 2013. – 436 с.

МОДЕЛИРОВАНИЕ СЕЙСМИЧЕСКИХ ПРОЦЕССОВ НА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

*Московский физико-технический институт, г. Долгопрудный,
k_h@inbox.ru*

Введение

Численное моделирование распространения динамических волновых возмущений в твердых телах используется при решении широкого круга задач. К таким задачам относятся задачи сейсморазведки, сейсмики и сейсмостойкости. Роль численного моделирования в каждой из данных областей очень важно. Численное моделирование распространения сейсмических волн представляет существенную часть работ при проведении геологоразведки в нефтяной отрасли. Математическое моделирование проводится в различных геологических средах, в том числе в слоистых средах и в средах с наличием различных неоднородностей (например трещины или каверны). Задачи такого рода представляются очень ресурсоемкими с точки зрения вычислительных ресурсов. Область вычисления, как правило, представляет собой сейсмический куб с длинной ребра от 1 км до 10 км. В то же время неоднородности могут быть размером в несколько метров. При моделировании задач сейсмостойкости также приходится сталкиваться с множеством неоднородностей. Параметры зданий, такие как толщина стен, размеры проемов, намного меньше размеров расчетных областей, включающих в себя порой большие массивы породы, размерами более 10 км вдоль одного направления. Таким образом, расчетная сетка должна быть достаточно подробной, чтобы иметь возможность правильно выделить все неоднородности. Для получения достаточной точности расчета и учета большого числа неоднородностей требуется использование больших вычислительных сеток, в реальных расчетах используются сетки размерами до нескольких десятков миллиардов узлов.

В данной работе рассматривается созданный программный комплекс для моделирования задач распространения динамических волновых возмущений в твердых телах. Комплекс работает на двумерных и трехмерных структурных блочных сетках с наличием неоднородностей. Для численного интегрирования применяются сеточно-характеристические и конечно-объемные методы 2-4 порядка точности.

Код распараллелен с использованием технологии MPI. Реализована низкоуровневая оптимизация кода с использованием потоковых

SIMD инструкций центральных процессоров SSE и AVX. В настоящее время достигнута эффективность распараллеливания до 70 %, используя технологию MPI при масштабировании до 16 тысяч вычислительных ядер. В системах с общей памятью алгоритм распараллелен, используя технологию OpenMP. Также код распараллелен, используя технологию CUDA, что дает ускорение до 50 раз по сравнению с одним ядром CPU. Программа может использовать несколько GPU устройств в рамках одного хоста.

В данной работе рассмотрены результаты распараллеливания одного и того же алгоритма, применяя различные технологии. Приведены тесты распараллеливания до 16 тысяч ядер CPU и 6 устройств CUDA. Приведены результаты тестовых расчетов.

Математическая модель и численный метод

Сформулируем основные уравнения линейной динамической теории упругости, которым подчиняется состояние бесконечно малого объема линейно-упругой среды. Рассмотрим нестационарные уравнения теории упругости для случая трех переменных в некоторой ортонормированной системе координат (x_1, x_2, x_3) :

$$\begin{aligned}\rho \cdot \dot{v}_i &= \nabla_j \cdot \sigma_{ij}, \\ \dot{\sigma}_{ij} &= q_{ijkl} \dot{\varepsilon}_{kl} + F_{ij}.\end{aligned}$$

Здесь ρ – плотность среды, v_i – компоненты вектора скорости смещения, σ_{ij} и ε_{kl} – компоненты тензоров напряжений Коши и деформации, ∇_j – ковариантная производная по j -й координате, F_{ij} – добавочная правая часть. Вид компонент тензора 4-го порядка q_{ijkl} определяется реологией среды. Для линейно-упругого случая они имеют вид

$$q_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}).$$

В этом соотношении, которое обобщает закон Гука, λ и μ – параметры Ламе, а δ_{ij} – символ Кронекера.

Также использовались сеточно-характеристические монотонные разностные схемы [1-2]. В программе реализованы схемы 2-4 порядка точности. Большинство расчетов проводилось с использованием схемы 3-го порядка точности.

Кроме того, используется сеточно-характеристический критерий монотонности, он имеет следующий вид:

$$\min \{u_m^n, u_{m-1}^n\} \leq u_m^{n+1} \leq \max \{u_m^n, u_{m-1}^n\}.$$

В простейшей реализации в случае невыполнения данного критерия монотонности применяется корректировка решения вида

$$u_m^{n+1} = \begin{cases} \max\{u_m^n, u_{m-1}^n\}, u_m^{n+1} > \max\{u_m^n, u_{m-1}^n\}, \\ \min\{u_m^n, u_{m-1}^n\}, u_m^{n+1} < \min\{u_m^n, u_{m-1}^n\}, \\ u_m^{n+1}, \min\{u_m^n, u_{m-1}^n\} \leq u_m^{n+1} \leq \max\{u_m^n, u_{m-1}^n\}. \end{cases}$$

Данный ограничитель сохраняет 3-й порядок в областях, где решение ведет себя достаточно гладко (выполняется характеристический критерий). В случае больших градиентов решения порядок схемы понижается до 2-го [2].

Программный комплекс

Для моделирования задач распространения волновых возмущений в гетерогенных средах был создан программный комплекс, позволяющий численно решать поставленные задачи, используя сеточно-характеристические методы разных порядков аппроксимации. При создании программного комплекса одним из основных требований была возможность проведения расчетов на достаточно больших расчетных сетках (более 10 млрд узлов), чтобы было возможно проводить моделирование в больших расчетных областях. В этом случае возникают значительные потребности как в ресурсах процессорного времени, так и в оперативной памяти ЭВМ. В работе рассмотрен ряд технологий, позволяющих существенно ускорить время работы вычислительного модуля.

Распараллеливание

Как уже упоминалось ранее, программный комплекс был распараллелен используя различные технологии. Алгоритм был оптимизирован путем задействования потоковых SIMD-расширений центрального процессора (SSE). SSE включает в архитектуру несколько расширенных регистров (8 на современных процессорах) размером 128 бит, либо 256 бит в случае технологии AVX. Преимущество в производительности достигается в том случае, когда необходимо произвести одну и ту же последовательность действий над разными данными. В таком случае блоком SSE осуществляется распараллеливание вычислительного процесса между данными. По результатам тестирования видно ускорение до семи раз при использовании чисел одинарной точности и до двух раз при числах с двойной точностью.

Для работы в системах с распределенной памятью программный комплекс распараллелен используя технологию MPI. При распараллеливании использовались стандартные алгоритмы декомпозиции расчетной области и обмена приграничными ячейками. Одно из основных требований к алгоритму – работа на большом числе вычислительных ядер (тысячи) для обеспечения приемлемого времени расчета. На рис. 1 приведены результаты тестирования масштабирования алгоритма при увеличении числа расчетных ядер от 128 до 16384.

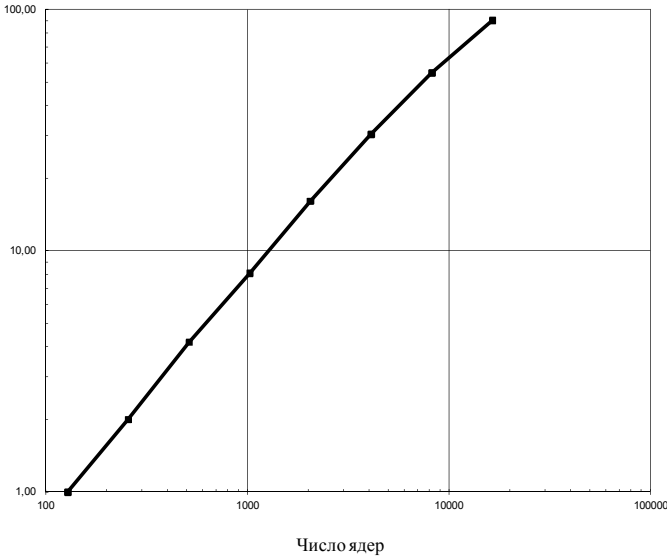


Рис. 1. Результаты тестирования распараллеливания MPI

Алгоритм также распараллелен в системах с общей памятью используя технологию OpenMP. В данном направлении работа только начата и при увеличении числа ядер от 1 до 32 алгоритм показывает эффективность около 60 %.

На графических GPGPU процессорах NVidia использовали технологию CUDA. Потребовалось полное переписывание части расчетного модуля под данную архитектуру. На текущий момент достигнуто ускорение около 80 раз по сравнению с одним ядром CPU, используя числа с одинарной точностью, и 30 раз, используя числа с двойной точностью.

Результаты расчетов

Рассмотрим процесс прохождения плоских продольной и поперечной волн через наземный объект, в нашем случае небольшое сооружение из бетона. Фронт волны параллелен свободной поверхности Земли, распространение волны от центра Земли к поверхности. В данном расчете такой импульс рассматривался как один пик от землетрясения. На рис. 2 показано прохождение продольной волны в различные моменты времени. Слева отображен модуль скорости среды, а справа цветом показаны участки, где хотя бы раз за расчет выполнилось условие Мизеса, т. е. возможны разрушения.

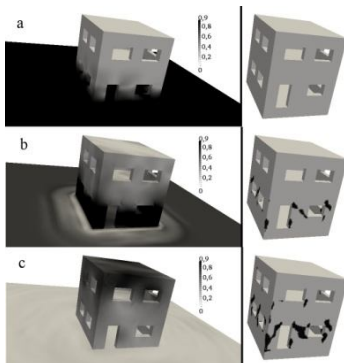


Рис. 2. Результаты расчета для продольной волны: а) $t = 0,0013$ с, б) $t = 0,0039$ с, в) $t = 0,0079$ с

1. Холодов А.С. О построении разностных схем повышенного порядка точности для уравнений гиперболического типа // Журнал вычислительной математики и математической физики, 1980. Т. 20, № 6. С. 1601 – 1620.
2. Холодов А.С., Холодов Я.А. О критериях монотонности разностных схем для уравнений гиперболического типа // Журнал вычислительной математики и математической физики, 2006. Т. 46, № 9. С. 1638 – 1667.

А.Б. Путилин, М.В. Степанов, Ж.В. Соколова

МАССИВНО ПАРАЛЛЕЛЬНЫЙ НЕЙРОСЕТЕВОЙ АЛГОРИТМ НА БАЗЕ CUDA В НАНОКОМПИЛЯТОРЕ ДЛЯ ОПТИМИЗАЦИИ ТОПОЛОГИИ 2D/3D НАНОСХЕМ НА БАЗЕ КВАНТОВЫХ КЛЕТОЧНЫХ АВТОМАТОВ

Московский государственный машиностроительный университет, г. Москва, putilin-45@yandex.ru, nanocompiler000@yandex.ru, sokolova.zh@yandex.ru

Физические основы и математический формализм описания переключения квантовых клеточных автоматов (ККА)

Для математического моделирования поведения и расчёта параметров наносхем на базе бестранзисторных вычислительных устройств (БТВУ) и оперативных запоминающих устройств (ОЗУ) на основе ККА-клапанов (рис.1, рис.2, www.nd.edu/~qcahome/) в нанокompиляторе (САПР НЭ: gtkwin_supercomputer2) [1 – 5] можно использовать аппарат искусственных нейронных сетей. Нейронные сети позволяют описать сложные взаимодействия между КТ и ККА с учётом 2D/3D топологии наносхем.

Для конструирования нейронной сети повторим необходимые физические основы работы наносхем на базе ККА. ККА (QCA) представляют

собой квадратную наноструктуру из 4-х или 5-ти КТ (QD). Кулоновское взаимодействие заставляет 2 электрона, имеющиеся в ККА, располагаться в диаметрально противоположных углах ККА. Таким образом, обеспечиваются два равноправных стабильных состояния поляризации ККА, одно из которых принимается за логический «0», а другое – за логическую «1». Физически состояние зарядовой поляризации определяется формулой

$$P = \frac{(\rho_1 + \rho_3) - (\rho_2 + \rho_4)}{\rho_0 + \rho_1 + \rho_2 + \rho_3 + \rho_4}, \quad (1)$$

где $\rho_i = \rho_i(t) = \rho(x, y, z, t)$ – плотность вероятности обнаружить электрон в i -той КТ, то есть, по всему i -му квадранту ККА $V_i = X_i \times Y_i \times Z_i$ в момент времени t ,

$$\begin{aligned} \rho_i(t) = \rho(x, y, z, t) &= \int_{X_i} \int_{Y_i} \int_{Z_i} |\Psi_0(x, y, z, t)|^2 dx dy dz = \\ &= \sum_{\sigma} \langle \Psi_0 | n_{i, \sigma} | \Psi_0 \rangle = |\Psi_0\rangle \langle \Psi_0| = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix} = \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{pmatrix}. \end{aligned} \quad (2)$$

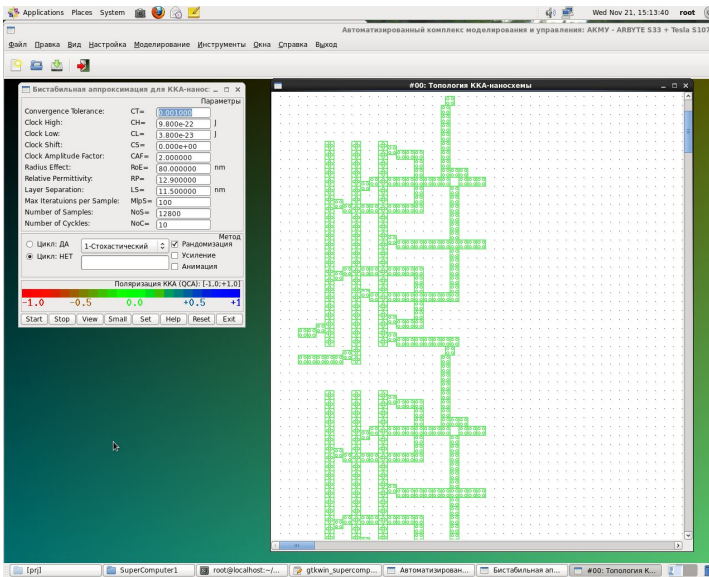
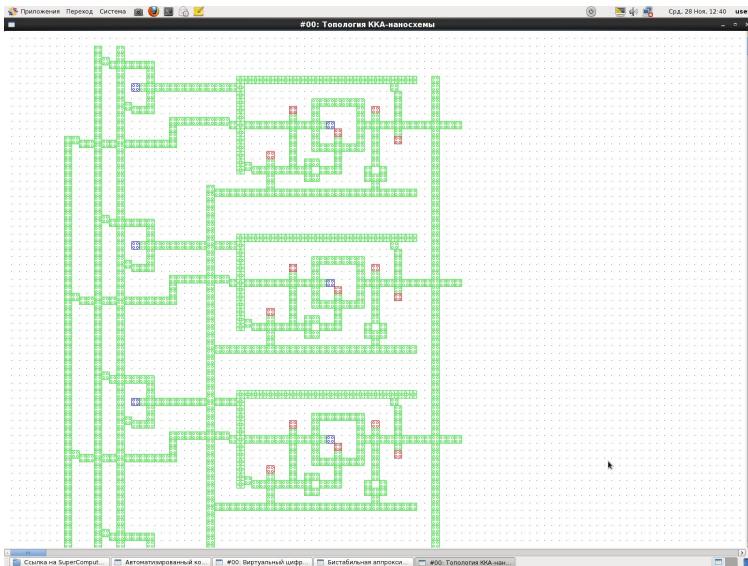


Рис. 1. Топология БТВУ на базе ККА-вентилей. 8-битный полный бинарный сумматор

Путём преобразований можно показать, что взаимовлияние поляризации двух соседних электростатически связанных ККА можно оценить по упрощённой формуле

$$P_i = \frac{\frac{E_{i,j}^k}{2\gamma_i} P_j}{\sqrt{1 + \left(\frac{E_{i,j}^k}{2\gamma_i} P_j \right)^2}}. \quad (3)$$

Значения энергий электронов γ_i , которые туннелируют внутри ККА можно оценить на основе информации о разрешённых уровнях энергии для квантовых ям, возникающих в КТ (QD) с заданными геометрическими характеристиками – высотой потенциального барьера, шириной. Для этого решается задача поиска собственных значений стационарного уравнения Шрёдингера для квантовой ямы, возникающей в данной КТ. Для простоты можно выбрать вариант прямоугольной квантовой ямы.



**Рис. 2. Топология ОЗУ на базе ККА-вентилей:
16-битный элемент памяти**

Описание вычисления состояний поляризации ККА на базе математического аппарата нейронных сетей

Формула (3) описывает зависимость значения поляризации i -го ККА от значений поляризации остальных ККА. Эту формулу можно использовать как основу для конструирования нейронной сети, адаптивно описывающей стационарное состояние наносхемы на базе ККА в зависимости от её 2D/3D топологии.

1. Нейронная сеть будет иметь 1 слой с полными последовательными связями и полными обратными связями как нейронная сеть Хопфилда. Выходным значением i -того нейрона является P_i – состояние поляризации i -го ККА. Полные последовательные связи каждого входа с каждым нейроном аппроксимируют электростатическое взаимодействие всех ККА в наносхеме. Сила такого взаимодействия убывает как квадрат расстояния между ККА.

2. Весовые коэффициенты синапсов имеют фиксированные неизменяемые значения. Это значения кинк-энергии $E_{i,j}^k = E_k$ – энергия взаимодействия (перекручивания – kink-energy) между i тым ККА и j -тым ККА, которая ассоциируется с энергетической ценой переключения поляризации ККА. Эти значения вычисляются для каждой 2D/3D топологии наносхемы на основе закона Кулона.

3. Функционал энергии ошибки нейронной сети выводится в следующем виде:

$$E_i^{(k)} = \left[P_i^{(k)} - \frac{a \sum_{j \neq i} E_{ij}^{kink} P_j^{(k)}}{\sqrt{1 + \left(a \sum_{j \neq i} E_{ij}^{kink} P_j^{(k)} \right)^2}} \right]^2, \quad (4)$$

$$E^{(k)} = \frac{1}{2} \sum_{i=0}^{N-1} E_i^{(k)} \quad (5)$$

4. Для оптимизации состояния нейронной сети используем метод градиентного спуска:

$$P_i^{(k+1)} = P_i^{(k)} + \alpha \cdot \Delta P_i^{(k)}; \quad \alpha \cong 0,001, \quad (6)$$

$$\Delta P_i^{(k)} = -\frac{\partial E^{(k)}}{\partial P_i^{(k)}} = -\left[\frac{\partial E_i^{(k)}}{\partial P_i^{(k)}} + \frac{\partial}{\partial P_i^{(k)}} \sum_{j \neq i} E_j^{(k)} \right] = -[A + B], \quad (7)$$

$$A = \frac{\partial E_i^{(k)}}{\partial P_i^{(k)}} = \left[P_i^{(k)} - \frac{a \sum_{j \neq i} E_{ij}^{kink} P_j^{(k)}}{\sqrt{1 + \left(a \sum_{j \neq i} E_{ij}^{kink} P_j^{(k)} \right)^2}} \right], \quad (8)$$

$$B = \frac{\partial}{\partial P_i^{(k)}} \sum_{j \neq i} E_j^{(k)} = \frac{\partial}{\partial P_i^{(k)}} \sum_{j \neq i} \left[P_j^{(k)} - \frac{a \sum_{j \neq i} E_{ij}^{kink} P_j^{(k)}}{\sqrt{1 + \left(a \sum_{j \neq i} E_{ij}^{kink} P_j^{(k)} \right)^2}} \right]. \quad (9)$$

5. Формулы (6 – 9) описывают алгоритм настройки нейронной сети. При сходимости нейронной сети получается решение, показывающее стационарное состояние поляризаций ККА наносхемы на базе ККА.

Описание самоорганизации наносхем на основе ККА на базе математического аппарата нейронных сетей

Для математического моделирования поведения таких сложных систем как наносхемы на базе ККА можно использовать аппарат искусственных нейронных сетей. Нейронные сети позволят описать сложные взаимодействия между КТ и ККА с учётом 2D/3D топологии наносхем.

1. Каждый ККА – маленькая однослойная нейронная сеть с полными последовательными связями (каждый-вход-с-каждым-нейроном) и обратными связями, которая имеет 4 входа и 4 нейрона – среднее между сетью Хопфилда и Кохонена.

2. Весовые коэффициенты синапсов маленькой однослойной нейронной сети ККА жёстко задаются 2D топологией ККА (ориентация 90° или 45°) и законом Кулона для 2-х электронов в ККА.

3. Каждый нейрон нейронной сети, аппроксимирующей работу наносхемы на базе ККА, отображает поведение отдельной КТ внутри ККА. Каждый нейрон связан со всеми другими нейронами нейронной сети. Весовые коэффициенты синапсов определяются на основе закона Кулона – как кинк-энергия, только не для всей ККА, а для каждой КТ внутри ККА. Таким образом, можно учесть конкретную 2D топологию наносхемы на базе ККА со сдвигами отдельных ККА, ориентацией 90° или 45°.

4. Поляризация ККА определяется по формуле через плотность зарядов в КТ внутри ККА (нумерация начинается с 0, для $N = 4$: $i = [0; 1; 2; 3]$):

$$P_{QCA} = \frac{(p_0 + p_2) - (p_1 + p_3)}{p_0 + p_1 + p_2 + p_3}; \quad (10)$$

$$p_i \in [0, 0; 1, 0]; \quad P_{QCA} \in [-1, 0; +1, 0].$$

5. Каждой КТ, которых в каждом ККА 4 штуки, поставим в соответствие один нейрон, выходной сигнал которого будет аппроксимировать значение плотности вероятности обнаружить электрон в КТ внутри ККА — $p_i^{(k+1)}$. Тогда для каждой КТ можно определить влияние, которое на электрон внутри неё оказывают посредством Кулоновского отталкивания (Кулоновской блокады) другие электроны из других КТ, как того же ККА, так и других ККА наносхемы. Очевидно, что такое влияние зависит от 2D/3D топологии наносхемы и обратно пропорционально расстоянию между КТ r_{ij} . Итак, для каждого нейрона можно написать следующую формулу, учитывающую самоорганизацию внутри каждого ККА по типу самоорганизующейся нейросети Кохонена:

$$p_i^{(k+1)} = f_1 \left(p_i^{(k)} + \alpha \cdot \Delta p_i^{(k)} \right) = f_1 \left(p_i^{(k)} - \frac{\partial W_{QCA}^{(k)}}{\partial p_i^{(k)}} \right), \quad (11)$$

$$f_1(x) = \frac{1}{1 + \exp(-\beta \cdot x)}; \quad \alpha \cong 0,001; \quad \beta \cong 100,0.$$

6. Функционал энергии ошибки для отдельного ККА можно сконструировать на основе формулы потенциальной энергии для ансамбля одноимённых точечных зарядов (электронов):

$$W_{QCA} = \sum_{i=0}^3 W_i = \sum_{i=0}^3 \sum_{j=0}^3 W_{ij} =$$

$$= \sum_{i=0}^3 \left[\frac{q_i}{4\pi\epsilon\epsilon_0} \sum_{j=0}^3 \frac{q_j}{r_{ij}} \right] = \frac{1}{4\pi\epsilon\epsilon_0} \sum_{i=0}^3 \left[q_i \sum_{j=0}^3 \frac{q_j}{r_{ij}} \right]. \quad (12)$$

7. Для плотностей вероятности обнаружить электрон в КТ внутри ККА можно ввести эквивалентное выражение без учёта влияния КТ из других ККА наносхемы:

$$W_{QCA} = \sum_{i=0}^3 \left[p_i \sum_{j=0}^3 \frac{p_j}{r_{ij}} \right] =$$

$$= 2 \cdot \left[\frac{p_0 p_1}{r_{01}} + \frac{p_0 p_2}{r_{02}} + \frac{p_0 p_3}{r_{03}} + \frac{p_1 p_2}{r_{12}} + \frac{p_1 p_3}{r_{13}} + \frac{p_2 p_3}{r_{23}} \right]. \quad (13)$$

8. Для оптимизации состояния маленькой однослойной нейронной сети внутри ККА, без учёта влияния КТ из других ККА наносхемы, используем метод градиентного спуска:

$$p_i^{(k+1)} = p_i^{(k)} + \alpha \cdot \Delta p_i^{(k)}; \quad \alpha \cong 0,001, \quad (14)$$

$$\Delta p_i^{(k)} = - \frac{\partial W_{QCA}^{(k)}}{\partial p_i^{(k)}} = \frac{\partial}{\partial p_i^{(k)}} \left[\sum_{i=0}^3 \left[p_i^{(k)} \sum_{j=0}^3 \frac{p_j^{(k)}}{r_{ij}} \right] \right] = 2 \cdot \sum_{j=0}^3 \frac{p_j^{(k)}}{r_{ij}}, \quad (15)$$

$$\Delta p_0^{(k)} = - \frac{\partial W_{QCA}^{(k)}}{\partial p_0^{(k)}} = -2 \cdot \left[\frac{p_1^{(k)}}{r_{01}} + \frac{p_2^{(k)}}{r_{02}} + \frac{p_3^{(k)}}{r_{03}} \right], \quad (16)$$

$$\Delta p_1^{(k)} = - \frac{\partial W_{QCA}^{(k)}}{\partial p_1^{(k)}} = -2 \cdot \left[\frac{p_0^{(k)}}{r_{01}} + \frac{p_2^{(k)}}{r_{12}} + \frac{p_3^{(k)}}{r_{13}} \right], \quad (17)$$

$$\Delta p_2^{(k)} = - \frac{\partial W_{QCA}^{(k)}}{\partial p_2^{(k)}} = -2 \cdot \left[\frac{p_0^{(k)}}{r_{02}} + \frac{p_1^{(k)}}{r_{12}} + \frac{p_3^{(k)}}{r_{23}} \right], \quad (18)$$

$$\Delta p_3^{(k)} = - \frac{\partial W_{QCA}^{(k)}}{\partial p_3^{(k)}} = -2 \cdot \left[\frac{p_0^{(k)}}{r_{03}} + \frac{p_1^{(k)}}{r_{13}} + \frac{p_2^{(k)}}{r_{23}} \right], \quad (19)$$

9. Для учёта влияния КТ из других ККА добавим второе слагаемое. Первое аппроксимирует взаимоотталкивание внутри ККА, второе аппроксимирует влияние на каждую КТ других КТ из других (N—1) ККА, входящих в наносхему с заданной 2D/3D топологией из N штук ККА. Каждый ККА даёт общей нейронной сети, аппроксимирующей работу наносхемы, 4 нейрона, соответствующих 4-м КТ внутри ККА.

$$\Delta p_i^{(k)} = - \left[\frac{\partial W_{QCA}^{(k)}}{\partial p_i^{(k)}} + f_2 \left(\sum_{j=0}^{N-1} p_j^{(k)} \cdot E_{ij}^{kink} \right) \right], \quad (20)$$

$$f_2(y) = \frac{1}{1 + \exp(-\gamma \cdot y)}; \quad \gamma \cong 100,0,$$

$$\Delta p_0^{(k)} = - \frac{\partial W_{QCA}^{(k)}}{\partial p_0^{(k)}} = - \left[2 \cdot \left[\frac{p_1^{(k)}}{r_{01}} + \frac{p_2^{(k)}}{r_{02}} + \frac{p_3^{(k)}}{r_{03}} \right] + f_2 \left(\sum_{j=0}^{N-1} p_j^{(k)} \cdot E_{ij}^{kink} \right) \right], \quad (21)$$

$$\Delta p_1^{(k)} = - \frac{\partial W_{QCA}^{(k)}}{\partial p_1^{(k)}} = - \left[2 \cdot \left[\frac{p_0^{(k)}}{r_{01}} + \frac{p_2^{(k)}}{r_{12}} + \frac{p_3^{(k)}}{r_{13}} \right] + f_2 \left(\sum_{j=0}^{N-1} p_j^{(k)} \cdot E_{ij}^{kink} \right) \right], \quad (22)$$

$$\Delta p_2^{(k)} = - \frac{\partial W_{QCA}^{(k)}}{\partial p_2^{(k)}} = - \left[2 \cdot \left[\frac{p_0^{(k)}}{r_{02}} + \frac{p_1^{(k)}}{r_{12}} + \frac{p_3^{(k)}}{r_{23}} \right] + f_2 \left(\sum_{j=0}^{N-1} p_j^{(k)} \cdot E_{ij}^{kink} \right) \right], \quad (23)$$

$$\Delta p_3^{(k)} = - \frac{\partial W_{QCA}^{(k)}}{\partial p_3^{(k)}} = - \left[2 \cdot \left[\frac{p_0^{(k)}}{r_{03}} + \frac{p_1^{(k)}}{r_{13}} + \frac{p_2^{(k)}}{r_{23}} \right] + f_2 \left(\sum_{j=0}^{N-1} p_j^{(k)} \cdot E_{ij}^{kink} \right) \right]. \quad (24)$$

10. Итак, на основе (11) и (21 – 24) составим уравнение настройки нейрона нейронной сети, аппроксимирующей состояние наносхемы на основе 2D/3D топологии расположения ККА:

$$\begin{aligned} p_i^{(k+1)} &= f_1 \left(p_i^{(k)} + \alpha \cdot \Delta p_i^{(k)} \right) = \\ &= f_1 \left(p_i^{(k)} - \alpha \cdot \left[2 \cdot \left[\frac{p_{i-3}^{(k)}}{r_{(i-3)i}} + \frac{p_{i-2}^{(k)}}{r_{(i-2)i}} + \frac{p_{i-1}^{(k)}}{r_{(i-1)i}} \right] + f_2 \left(\sum_{j=0}^{N-1} p_j^{(k)} \cdot E_{ij}^{kink} \right) \right] \right) \end{aligned} \quad (25)$$

пример для $i = 3$.

Распараллеливание по технологии NVIDIA CUDA по взвешенным суммам нейронов

Технология массивно параллельной обработки информации для численного моделирования NVIDIA CUDA на многоядерных устройствах GPU (Graphic Processor Unit: 240 kernel) и Tesla S1070/S1070-500 (4×GPU: 960 kernel) позволяет организовать распараллеливание задачи (процесса) моделирования, разделив его на 960 или 1920 (для 2-х Tesla S1070) параллельных вычислительных потоков (нитей). Каждый такой

вычислительный поток (нить) будет привязан к своему отдельному физическому ядру (kernel) внутри GPU. Все эти вычислительные потоки (нити) работают физически параллельно друг с другом.

1. За каждым ККА закрепляется нить и соответствующее ядро. В этом случае общее число нитей равно числу ККА в наносхеме.
 2. За каждой КТ в ККА закрепляется нить и соответствующее ядро. В этом случае общее число нитей равно числу ККА в наносхеме, помноженному на 4.
1. Дубовой А.Н., Панченков И.В., Пушкарский С.В., Егоров С.Д., Мазур С.Н., Пустовалов Е.В., Соколова Ж.В., Степанов М.В. Многокластерная нанотехнологическая установка для исследования и изготовления наносхем и функциональных наносистем для систем энергообеспечения космических аппаратов // V Белорусский космический конгресс, 25—27.10.2011, Минск.
 2. Дубовой А.Н., Родионов Б.Н., Егоров С.Д., Мазур С.Н., Коротеев А.В., Соколова Ж.В., Степанов М.В., Хлюстов П.М. Перспективы применения многокластерной нанотехнологической установки для изготовления nano-элементов энергосистем космических аппаратов // Доклады и Труды Международной конференции «Нанотехнология-производству-2013», Фрязино, апрель 2013.
 3. Дубовой А.Н., Родионов Б.Н., Егоров С.Д., Мазур С.Н., Коротеев А.В., Степанов М.В., Хлюстов П.М., Многокластерная нанотехнологическая установка для исследования и изготовления наносхем и функциональных наносистем для систем энергообеспечения космических аппаратов // Нанотехника №1(33), 2013.
 4. Путилин А.Б. Континуальные системы обработки информации. Издательство «Квадрат С», Москва, 2005.
 5. Путилин А.Б. Физические системы преобразования и обработки информации. Основы построения континуальных систем преобразования и обработки информации. Издательство МГОУ, Москва, 2012.

М.Д. Чекина

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЗАДАЧ ГЕОФИЛЬТРАЦИИ В ПОЧВОГРУНТАХ С ФРАКТАЛЬНОЙ СТРУКТУРОЙ НА МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

*Научно-образовательный центр «Комплексные исследования
и математическое моделирование» НИИ многопроцессорных
вычислительных систем ЮФУ, г. Таганрог
elfik55@gmail.com*

При изучении многих физических явлений приходится иметь дело с движением жидкостей в пористых средах – фильтрацией. В

таких фильтрационных процессах, примерами которых могут служить просачивание воды через почву, движение нефти в подземных пластах и т.п., жидкость движется по разветвленной системе сообщающихся между собой пор. В данной работе для моделирования процессов фильтрации в пористых средах используется уравнение Букингема-Ричардса [1]

$$C \cdot D_{[0,t]}^{\chi} h = (k(h)h_x')_x' + (k(h)h_y')_y' + (k(h)h_z')_z' + \tilde{Q}. \quad (1)$$

В данном уравнении x, y – горизонтальные координаты, z – вертикальная, θ – влажность, h – потенциал почвенной влаги, $k(h)$ – коэффициент влагопроводности, z_0 – толщина распределения источников и потерь, F – источники (приток, осадки и др.), Q – потери (сток, испарение и др.), $D_{[0,t]}^{\chi} f(t) = \frac{1}{\Gamma(1-\chi)} \int_0^t \frac{f(\tau) d\tau}{(t-\tau)^{\chi}}$ – регуляризованная дробная производная Римана-Луивилля порядка χ ($0 < \chi < 1$).

Зависимость коэффициента влагопроводности $k(h)$ от потенциала h имеет вид

$$\begin{cases} k(h) = k_f e^{\alpha h}, & h < 0 \\ k(h) = k_f, & h \geq 0, \end{cases}$$

где k_f – коэффициент фильтрации, α – эмпирический параметр.

Для влажности θ будет справедлива следующая зависимость от потенциала h :

$$\begin{cases} \theta(h) = (\theta_s - \theta_r) e^{\alpha h} + \theta_r, & h < 0 \\ \theta(h) = \theta_s, & h \geq 0, \end{cases}$$

где θ_s – максимальная влажность, θ_r – минимальная влажность.

Дополним уравнение (1) граничными условиями, для этого воспользуемся законом Дарси:

$$v = -k \text{grad}(h), \quad (2)$$

где v – скорость фильтрации флюида.

В случае непроницаемой стенки должно выполняться требование отсутствия потока флюида через эту границу $v \cdot \mathbf{n} = 0$. Это требование в совокупности с законом Дарси (2) дает следующее граничное условие:

$$\frac{\partial h}{\partial \mathbf{n}} = 0.$$

В случае если через границу задан поток жидкости S , то, исходя из тех же соображений, будем иметь следующее граничное условие:

$$\frac{\partial h}{\partial \mathbf{n}} = S.$$

Во многих случаях логично предположить, что поток жидкости зависит от потенциала линейно, тогда получим следующее граничное условие:

$$\frac{\partial h}{\partial \mathbf{n}} = \alpha h + \beta.$$

С помощью интегро-интерполяционного метода строим разностный аналог уравнения (1) на неравномерной сетке

$$\bar{\omega} = \left\{ (x_i, y_j, z_k), x_i = i \cdot h_x, y_j = j \cdot h_y, z_k = k \cdot h_z, i = 1..n_1, j = 1..n_2, k = 1..n_3 \right\}.$$

Дробную производную $D_{[-\infty, t]}^\chi h$ заменим дискретным аналогом [2]:

$$D_{[-\infty, t]}^\chi h \square \frac{1}{\Gamma(2-\chi)} \sum_{s=1}^n (t_{n-s+1}^{1-\chi} - t_{n-s}^{1-\chi}) h_{t, i, j, k}^s.$$

Запишем полученный дискретный аналог уравнения (1) в каноническом виде:

$$\begin{aligned} h_{i, j, k}^{n+1} & \left(\frac{C_{i, j, k}}{\Gamma(2-\chi)\tau^\chi} + \frac{k \frac{i+\frac{1}{2}, j, k}{h_x^2}}{h_x^2} + \frac{k \frac{i-\frac{1}{2}, j, k}{h_x^2}}{h_x^2} + \frac{k \frac{i, j+\frac{1}{2}, k}{h_y^2}}{h_y^2} + \frac{k \frac{i, j-\frac{1}{2}, k}{h_y^2}}{h_y^2} + \frac{k \frac{i, j, k+\frac{1}{2}}{h_z^2}}{h_z^2} + \frac{k \frac{i, j, k-\frac{1}{2}}{h_z^2}}{h_z^2} \right) - \\ & - h_{i+1, j, k}^{n+1} \frac{k \frac{i+\frac{1}{2}, j, k}{h_x^2}}{h_x^2} - h_{i-1, j, k}^{n+1} \frac{k \frac{i-\frac{1}{2}, j, k}{h_x^2}}{h_x^2} - h_{i, j+1, k}^{n+1} \frac{k \frac{i, j+\frac{1}{2}, k}{h_y^2}}{h_y^2} - h_{i, j-1, k}^{n+1} \frac{k \frac{i, j-\frac{1}{2}, k}{h_y^2}}{h_y^2} - \\ & - h_{i, j, k+1}^{n+1} \frac{k \frac{i, j, k+\frac{1}{2}}{h_z^2}}{h_z^2} - h_{i, j, k-1}^{n+1} \frac{k \frac{i, j, k-\frac{1}{2}}{h_z^2}}{h_z^2} = C_{i, j, k} \frac{2-2^{1-\chi}}{\Gamma(2-\chi)\tau^\chi} h_{i, j, k}^n + \\ & + \frac{C_{i, j, k}}{\Gamma(2-\chi)\tau} \left[(t_{n+1}^{1-\chi} - t_n^{1-\chi}) h_{i, j, k}^0 + (-t_{n+1}^{1-\chi} + 2t_n^{1-\chi} - t_{n-1}^{1-\chi}) h_{i, j, k}^1 + \right. \\ & \left. + \dots + (-t_3^{1-\chi} + 2t_2^{1-\chi} - t_1^{1-\chi}) h_{i, j, k}^{n-1} \right] + k_{i, j, k} + Q_{i, j, k} \end{aligned}$$

Для повышения оперативности получения результата моделирования целесообразно распараллеливать вычисления на сетках с большим количеством ячеек.

Параллельная версия программного комплекса разработана на языке высокого уровня C++ с использованием библиотеки передачи сообщений MPI.

Так как для расчета мы используем метод верхней релаксации, то для того чтобы посчитать значение в (i, j) узле, нам необходимо знать значения в $(i-1, j)$, $(i+1, j)$, $(i, j-1)$, $(i, j+1)$ узлах. Для этого пер-

вому и последнему процессам мы передаем на один столбец больше, а остальным процессам мы передаем на два столбца больше. Эти столбцы передаются только на чтения, расчеты в них не производятся.

На каждом процессе вычисляется невязка. Затем каждый процесс пересылает свою невязку на нулевой процесс, на котором вычисляется максимальная невязка, которая затем сравнивается с погрешностью ε .

В нашем случае распараллеливание задачи затрудняется необходимостью использования динамических масок для определения расчетных и нерасчетных узлов сетки. Поэтому после расчета каждого нового слоя воды матрица собирается на нулевом процессе, который является управляющим в параллельной реализации, для того чтобы обновить маску и выявить новые расчетные и нерасчетные узлы. Затем вновь происходит расчет заполнения области водой и выполняется необходимое количество итераций.

На рис. 1 и рис. 2 приведены графики зависимости эффективности (E) от количества процессоров, выполняющих заданную задачу. Расчеты проводились для областей с размерами 16×16 , 32×32 , 64×64 , 128×128 , 256×256 , 512×512 ячеек сетки.

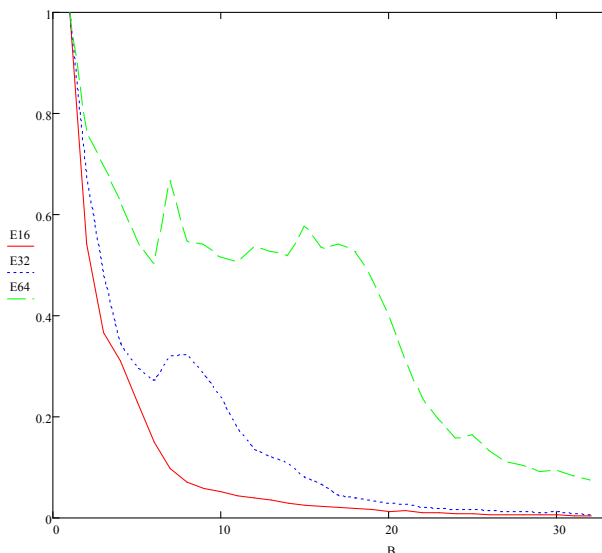


Рис. 1. Графики зависимости эффективности от количества процессоров для областей с размерами 16×16 , 32×32 , 64×64

Из графиков видно, что эффективность работы программы растет с увеличением числа ячеек в сетке. Для сеток с небольшим количеством узлов (16×16 , 32×32) нецелесообразно использовать количество процессоров больше двух. В расчетах на сетках с менее чем 2^{11} ячейками следует задей-

ствовать не более 10-ти процессорных элементов, а при количестве ячеек около 2^{12-14} (области 64×64 и 128×128) возможно использование до 20 процессоров. Если же расчетная сетка состоит из 2^{16-18} ячеек, то вполне целесообразно использование около 30 вычислительных элементов. Падение эффективности после определенного порога объясняется тем, что время, затрачиваемое на прием-передачу данных, а также сборку-разборку области становится слишком большим по сравнению со временем вычисления.

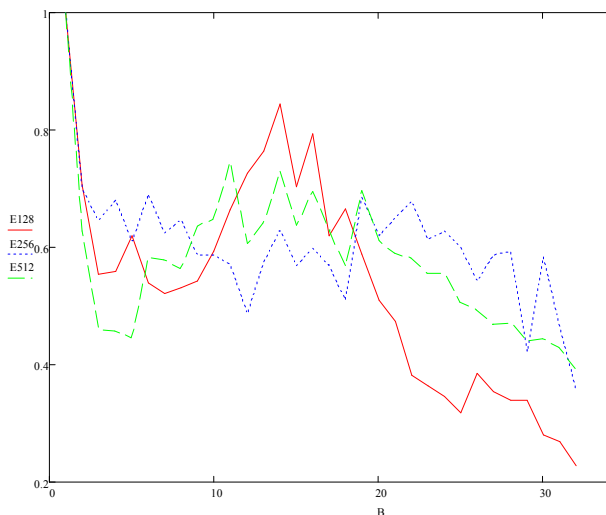


Рис.2. Графики зависимости эффективности от количества процессоров для областей с размерами 128x128, 256x256, 512x512

Итак, для целесообразности использования в работе супервычислительных систем следует брать сетки с количеством узлов более чем 2000 для количества процессоров более десяти.

1. Кашенко Н.М., Никитин М.А. Моделирование эффектов аномальной диффузии для дренажных схем.
2. Шхануков-Лафишев М.Х., Такенова Ф.И. Разностные методы решения краевых задач для дифференциальных уравнений дробного порядка // Журнал вычислительной математики и математической физики. 2006. Том 46, №10. С. 1871-1881.
3. Самарский А.А. Теория разностных схем. М.: Наука. 1977.
4. Сухинов А.И., Чекина М.Д. Математическая модель и численный метод для задачи динамики выпадения осадков и затопления // Известия ЮФУ. Технические науки, №8, 2009. С. 42-52.

РАСЧЕТ НА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ РАСПРОСТРАНЕНИЯ В ВОДОЕМЕ НЕФТЯНОГО ПЯТНА *

*Южно-Российский региональный центр информатизации ЮФУ,
г. Ростов-на-Дону,
shabas@sfedu.ru*

В работе рассматривается процесс распространения нефтяного пятна, попавшего в водоем. Нефтяные загрязнения – это неизбежная опасность, возникающая в процессе транспортировки нефти водным путем от мест ее добычи к местам переработки. Нефтяное загрязнение оказывает негативное влияние на все процессы, протекающие в природных водоемах и на их границах с атмосферой и сушей. Поэтому возникает необходимость прогнозирования поведения нефти, попавшей по той или иной причине в акваторию водоема. Одним из путей подобного прогнозирования может стать математическое моделирование процесса распространения нефтяного загрязнения как на поверхности водоема, так и в его толще и на его границах.

При моделировании распространения нефти в водоеме необходимо учитывать большое количество разнообразных факторов. На поведение нефти, попавшей в водоем, оказывают влияние и собственные физико-химические свойства, и внешние условия окружающей среды. На первом этапе распространения преобладают процессы растекания нефтяного пятна. Параллельно с этими процессами под воздействием внешних природных факторов происходит неизбежная деструкция нефти. К процессу деструкции нефти относят испарение легких нефтяных фракций, их растворение в водах водоема, эмульсификация, биodeградация нефти и т.п. Каждый из этих процессов приводит к значительной потере массы вылившейся нефти, а также оказывает влияние на плотность и вязкость нефти. Тяжелые фракции нефти могут оказаться плотнее воды и осесть на дно водоема.

Как было показано в [1, 2], на этапе растекания нефти по поверхности водоема наблюдают три режима: инерционный, гравитационно-вязкий и режим поверхностного натяжения. Переход от инерционного к гравитационно-вязкому режиму происходит тогда, когда толщина пленки h становится достаточно малой, а толщина вязкого пограничного слоя δ достаточно большой, так что выполняется равенство $\delta = \sqrt{\nu \cdot t}$, где t – время с момента пролива нефти, ν – вязкость. Сила поверхностного натяжения становится существенной, когда она по порядку вели-

* Работа выполнена при финансовой поддержке Министерства образования и науки РФ (проект №1420, государственное задание ВУЗов, базовая часть).

чины равняется с градиентом давления. На определенном этапе поверхностное натяжение меняет знак и растекание прекращается.

Помимо растекания пятна нефти относительно его центра тяжести, важным является вопрос о дрейфе пятна нефти. Поведение пятна нефти на поверхности водоема будем описывать двумерным уравнением конвекции – диффузии [3]:

$$\begin{aligned} \frac{\partial h}{\partial t} + \bar{\nabla}(h\bar{v}) - \bar{\nabla}(D\bar{\nabla}h) &= R_h, \\ \bar{v} &= \left(u_x + \frac{\tau_x^w}{c_f}, u_y + \frac{\tau_y^w}{c_f}\right), \\ D &= \frac{gh^2(\rho_w - \rho_{oil})}{\rho_w C_f}, \end{aligned} \quad (1)$$

где h – толщина нефти, $\bar{v}$ – скорость дрейфа пленки, $\frac{\tau_x^w}{c_f}$ – напряжение сдвига из-за ветра, D – функция диффузии распространения пятна нефти, C_f – коэффициент трения между нефтяной пленкой и поверхностью воды (0.02 кг/м²с), R_h – источниковая (стоковая) функция, g – ускорение свободного падения, $\bar{\nabla} = (\partial/\partial x, \partial/\partial y)$.

Распространение растворенных фракций нефти в толще водоема моделируется трехмерным уравнением конвекции – диффузии [44]:

$$\begin{aligned} \frac{\partial S}{\partial t} + u \frac{\partial S}{\partial x} + v \frac{\partial S}{\partial y} + w \frac{\partial S}{\partial z} = \\ = \frac{\partial}{\partial x} \left(\mu_{xy}(x, y) \frac{\partial S}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu_{xy}(x, y) \frac{\partial S}{\partial y} \right) + \frac{\partial}{\partial z} \left(\mu_z(z) \frac{\partial S}{\partial z} \right) + f(x, y, z), \end{aligned} \quad (2)$$

где $S(x, y, z)$ – концентрация загрязнения; u, v, w – составляющие вектора скорости по направлениям соответственно x, y, z ; $\mu_{xy}(x, y)$, $\mu_z(z)$ – коэффициенты горизонтальной и вертикальной турбулентной диффузии; $f(x, y, z)$ – функция источника.

Полученные системы уравнений решаются конечно-разностными методами с использованием неявных схем. В расчетной области строится прямоугольная равномерная по всем направлениям сетка. Для аппроксимации уравнений используется противопотоковая схема записи конвективных членов. В результате конечно-разностной аппроксимации получаются системы линейных уравнений с пятидиагональной матрицей (уравнение 1) и семидиагональной матрицей (уравнение 2).

Предполагается, что поле скоростей известно на каждом временном шаге.

Для проверки работоспособности задачи проводились численные эксперименты с изменением координат залповых источников выбросов нефти, с изменением количества нефти, попавшей в водоем в результате залпового выброса. Расчеты проводились при разных ветровых ситуациях.

Расчеты проводились на вычислительных кластерах ЮГИНФО ЮФУ с распределенной памятью в среде параллельного программирования MPI:

- **INFINI** – Linux-кластер, состоящий из 20 вычислительных узлов, соединенных скоростной коммуникационной сетью SDR Infiniband. Каждый вычислительный узел представляет собой компьютер с процессором Intel Pentium 4 3.4 ГГц и оперативной памятью DDR2 2Гб.
- **IBMX** – Linux-кластер, состоящий из 13 вычислительных узлов, соединенных скоростной коммуникационной сетью DDR Infiniband. Каждый вычислительный узел представляет собой компьютер с одним 2-ядерным процессором Intel Xeon 5160 с тактовой частотой 3.0 ГГц и оперативной памятью 8Гбайт.
- **LINUX** – вычислительный кластер из 10 узлов, соединенных вычислительной сетью Gigabit Ethernet. Каждый из узлов представляет собой компьютер с процессором Pentium 4 2.4 ГГц, с 512 Мб оперативной памяти.
- **WSD** – кластер из 8-ми рабочих станций DELL с 2-ядерными процессорами Intel Core 2 Duo, оперативной памятью 4 Гб и коммуникационной сетью Gigabit Ethernet.

Для решения систем линейных алгебраических уравнений, возникающих при дискретизации исходных дифференциальных уравнений, использовалась библиотека параллельных подпрограмм Aztec. В этой библиотеке реализован набор итерационных методов Крылова для решения систем линейных алгебраических уравнений с разреженными матрицами.

Результаты тестовых расчетов показали, что в расчетах имеет смысл применять метод CG из пакета Aztec, а расчеты на кластере WSD или IBMX обеспечивают лучшее время счета. Вычислительные эксперименты дают основание сделать вывод, что предложенная модель и ее программная реализация позволяют проводить расчеты, результаты которых достаточно хорошо согласуются с наблюдаемыми в природе процессами. Применение ВВС позволит в дальнейшем уточнять и усложнять предложенную модель для более реалистичного отражения ею реальных процессов.

1. J.A.Fay "The spread of oil slicks on a calm sea" In: Oil on the sea, Plenum Press. - New-York, 1969, p.53-63.
2. Fay J.A. "Physical processes in the spread of oil on a water surface" In: Proc. of h- o Joint Conf. on prevention and control of oil spills. Washington, 1971 (cit. N8).
3. Ehsan Sarhadi Zadeh1 and Kourosh Hejazi «Eulerian Oil Spills Model Using Finite-Volume Method with Moving Boundary and Wet-Dry Fronts» // Modeling and Simulation in Engineering Volume 2012 (2012), Article ID 398387, 7 pages, <http://dx.doi.org/10.1155/2012/398387>
4. Чикин А.Л., Шабас И.Н. Построение трехмерной гидрофизической модели Азовского моря // Изв. вузов, Сев.-Кав. регион. «Естественные науки», №3, 2001. С.33-37.

ПАРАЛЛЕЛЬНОЕ ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ЗАДАЧ ГИДРОДИНАМИКИ МЕЛКОВОДНЫХ ВОДОЕМОВ С УТОЧНЕННЫМИ ГРАНИЧНЫМИ УСЛОВИЯМИ

*Институт компьютерных технологий и информационной
безопасности ЮФУ, г. Таганрог,
shishenya.alexander@yandex.ru*

Введение

В современном мире водные объекты стали важнейшими природными ресурсами и поэтому подвергаются активному антропогенному воздействию, что оказывает воздействие на гидрологические режимы, температуру, состав воды и, как следствие, подвергает изменениям экосистему водоема, поэтому охрана и мониторинг состояния водных объектов является необходимой составляющей их эксплуатации.

В случае возникновения чрезвычайных ситуаций, связанных с выбросом опасных веществ в водоем, заморными явлениями, затоплением прибрежных районов, предпринятые в течении первых нескольких часов адекватные меры, основанные на прогнозе развития ЧС, позволят сократить материальный ущерб, и, что важнее, спасти человеческие жизни.

Следствием мелководности водоема является близкое расположение горизонтальных границ, поэтому неточности аппроксимации границы вносят более значимый вклад в конечный результат, чем при моделировании глубоководных водоемов. Таким образом, ставится задача построения программного комплекса для расчета задач гидродинамики мелководных водоемов с уточненными граничными условиями на параллельных вычислительных системах.

Формулировка непрерывной модели

В модель входят [1]:

уравнения движения (Навье–Стокса) в векторном виде:

$$(\rho \mathbf{v})'_t + \operatorname{div}(\rho \mathbf{v} \otimes \mathbf{v}) = -\operatorname{grad} P + \operatorname{div}(\eta \operatorname{grad} \mathbf{v}) + \rho \mathbf{f}, \quad (1)$$

уравнение неразрывности:

$$\rho'_t + \operatorname{div}(\rho \mathbf{v}) = 0, \quad (2)$$

уравнение, описывающее эволюцию функции заполнения:

$$q'_t + (\operatorname{grad} q)^T \cdot \mathbf{v} = |\operatorname{grad} q| \cdot \omega, \quad (3)$$

где η – тензор динамической вязкости, μ – тензор кинематической вязкости, ρ – функция плотности жидкости, $\mathbf{v}$ – вектор скорости движения среды, P – давление, $\mathbf{g}$ – вектор ускорения свободного падения, а $\mathbf{k}$ – сила Кориолиса, $\mathbf{f} = \mathbf{k} + \mathbf{g}$.

Граничные условия модели следуют из уравнений модели в предположении соблюдения физических свойств границы. В работе получено уточненное граничное условие для свободной поверхности.

Граничное условие для давления получено из уравнений (1) и (3):

$$\begin{aligned} |\mathbf{n}| \frac{\partial P}{\partial \mathbf{n}} = & -(\rho q'_t)' + (\rho \mathbf{n} \omega)'_t + \rho \mathbf{v} \cdot \mathbf{n}'_t - \\ & - \operatorname{div}(\rho \mathbf{v} \otimes \mathbf{v}) \cdot \mathbf{n} + \operatorname{div}(\eta \operatorname{grad} \mathbf{v}) \cdot \mathbf{n} + \rho \mathbf{f} \cdot \mathbf{n} \end{aligned}$$

Граничное условие для скорости учитывает касательные напряжения, вызываемые ветром и использует закон Ван-Дорна [2].

Условие для нахождения формы свободной поверхности:

$$q = \eta(P - P_a),$$

где P_a – атмосферное давление. Последнее граничное условие справедливо только при ограничении $\|\mathbf{v}'_t\| \ll g$.

Для непрерывной модели изучено влияние источников на суммарную механическую энергию системы, показано отсутствие нефизичных источников.

Дискретизация уравнений модели

Для аппроксимации уравнений модели по времени используется метод поправки к давлению. Также для улучшения аппроксимации границы для дискретизации пространственных производных используется интегро-интерполяционный метод, учитывающий частичную заполненность ячеек расчетной сетки. Приведем аппроксимацию конвективных и диффузионных слагаемых:

$$\begin{aligned} \iiint_{\Omega_{i,j,k}} (\rho u \varphi)'_x d\omega \cong & q_{i+1,j+\frac{1}{2},k+\frac{1}{2}} \frac{(\rho u \varphi)_{i+1,j,k} - (\rho u \varphi)_{i,j,k}}{2} h_y h_z + \\ & + q_{i,j+\frac{1}{2},k+\frac{1}{2}} \frac{(\rho u \varphi)_{i,j,k} - (\rho u \varphi)_{i,-1,j,k}}{2} h_y h_z, \\ \iiint_{\Omega_{i,j,k}} (\eta \varphi'_x)'_x d\omega \cong & q_{i+1,j+\frac{1}{2},k+\frac{1}{2}} \eta_{i+\frac{1}{2},j,k} \frac{\varphi_{i+1,j,k} - \varphi_{i,j,k}}{h_x} h_y h_z - \\ & - q_{i,j+\frac{1}{2},k+\frac{1}{2}} \eta_{i-\frac{1}{2},j,k} \frac{\varphi_{i,j,k} - \varphi_{i,-1,j,k}}{h_x} h_y h_z + \end{aligned}$$

$$+ \left(q_{i,j+\frac{1}{2},k+\frac{1}{2}} - q_{i+1,j+\frac{1}{2},k+\frac{1}{2}} \right) \eta_{i,j,k} \varphi'_{xi,j,k} h_y h_z,$$

где $\Omega_{i,j,k}$ – ячейка расчетной сетки.

Показано, что модель является консервативной, погрешность аппроксимации дискретной модели имеет второй порядок по пространству во внутренних узлах и первый в приграничных узлах. Также показано, что при использовании обычного интегро-интерполяционного метода в приграничных узлах имеет место нулевой порядок погрешности аппроксимации, то есть допускается константная ошибка.

Параллельная реализация расчетного алгоритма

Параллельная реализация программного комплекса выполнена на многопроцессорной высокопроизводительной системе НИИ МВС ЮФУ. Распараллеливание алгоритма выполнено методом декомпозиции области по двум пространственным направлениям, что позволяет увеличить эффективность использования параллельной системы.

Результаты расчетов

Для проверки правильности получаемых в результате работы программы результатов предлагается выполнить моделирование некоторых стандартных тестовых случаев. В качестве одной из таких модельных задач выбрана задача течения в скошенном резервуаре (рис. 1).

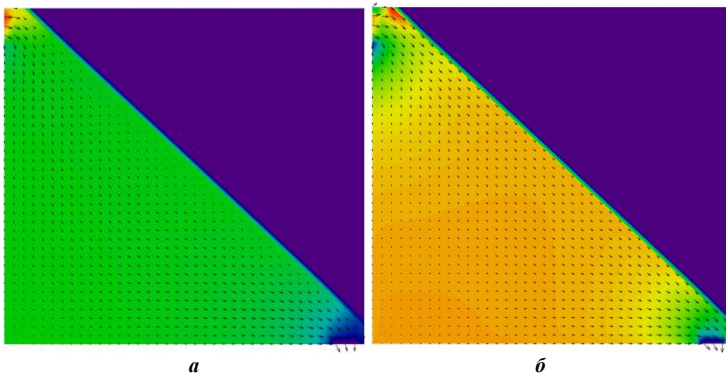


Рис.1. Результаты моделирования: а – без учета частичной заполненности ячеек расчетной сетки, б – с учетом частичной заполненности ячеек расчетной сетки. Векторное поле соответствует полю скорости, оттенками показано поле давления

Видно, что на рис. 1,а происходит потеря потока жидкости вдоль скошенной границы. Использование аппроксимации с учетом частичной заполненности и уточненных граничных условий позволяет избежать такого недостатка (рис.1,б). Также было выполнено математическое моделирование гидродинамики Таганрогского залива с использованием многопроцессорной вычислительной системы (рис. 2).

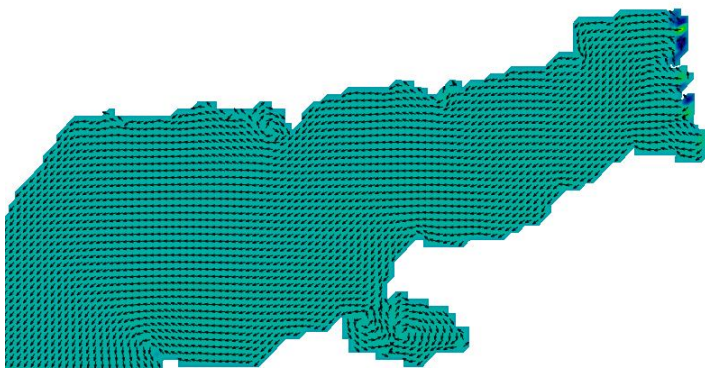


Рис. 2. Поля скорости и давления, полученные при моделировании гидродинамики Таганрогского залива. Скорость ветра полагается равной нулю

Заключение

В работе построена математическая модель гидродинамики мелководных водоемов с уточненными граничными условиями. Показано, что построенная модель обеспечивает более точное описание протекающего процесса в приграничной области, что, в некоторых случаях, приводит к значительному изменению картины течения. Выполнена программная реализация построенной модели, в том числе на многопроцессорной системе НИИ МВС ЮФУ.

1. Сухинов А.И., Чистяков А.Е., Тимофеева Е.Ф., Шишени А.В. Математическая модель расчета прибрежных волновых процессов // Матем. моделирование, 24:8 (2012), 32–44.
2. Васильев В.Сю, Сухинов А.И. Двумерные модели ветровых течений Азовского моря // Известия ЮФУ. Технические науки. Таганрог. Изд-во ТТИ ЮФУ, 2001, №2(20). С. 13-25.

АВТОРСКИЙ УКАЗАТЕЛЬ

А

Аксенов С.В., 231
Альбицкий Д.В., 145
Афонин А.А., 9

Б

Бахтерев М.О., 172
Безруков А.В., 177
Бкайтин А.В., 181
Бовкун А.В., 14
Богданова В.Г., 80
Боев С.Ф., 181
Бугаев С.А., 18
Буза М.К., 186
Бухановский А.В., 191, 197
Бычков И.В., 80
Бялик Ю.И., 70

В

Валуев И.А., 85
Васёв П.А., 172
Васильев В.Н., 191
Воронов А.А., 203
Вятчинин Д.А., 90

Г

Ганченко В.В., 203, 208
Голиков А.Н., 94
Горбунов В.С., 21
Горелова Г.В., 99
Гурьев Ю.В., 212

Д

Дацюк В.Н., 104
Дацюк О.В., 104
Дворкович А.В., 217
Дворкович В.П., 217
Доморацкий А.В., 90
Дордопуло А.И., 38
Дудкин А.А., 203, 208

Е

Егерев И.А., 109
Елизаров Г.С., 21

З

Завтур Е.Е., 18
Зачесов Ю.Л., 25
Змеев Д.Н., 222
Золотов И.Н., 25

И

Иванов Д.Я., 114, 141
Иванов С.В., 197
Иванова О.А., 160
Изгалин С.П., 240
Инютин А.В., 203

К

Каляев А.И., 118
Каляев И.А., 38
Катаев О.В., 29
Клименко А.Б., 99, 121
Климов А.В., 222
Коваленко А.Г., 55
Ковальчук С.В., 197
Колодзей А.В., 32
Кондратьева О.М., 186
Копытин А.С., 235
Коробкин В.В., 125
Коротков К.А., 35
Котляров А.С., 225
Кочкаров А.А., 181
Крукиер Л.А., 104, 226
Кузелин М.О., 70
Куповых В.Г., 129

Л

Лабошин Л.Ю., 132
Лайком Д.Н., 231
Левин И.И., 38
Левченко Н.Н., 222
Лукашин А.А., 132
Лукин Н.А., 46

М

Малахов И.Н., 21
 Маркович И.И., 235
 Марьев А.А., 235
 Машошин А.И., 136
 Мельник Э.В., 99, 141, 145
 Мельник Э.В.,
 Меркулов В.И., 149
 Мияков Д.А., 149
 Михайлин Д.А., 54
 Михеев В.А., 240
 Морозов И.В., 85
 Мурашко Н.И., 242

Н

Непочатых Е.В., 63, 67
 Нечаев Ю.И., 191, 197
 Николаева Е.Д., 90

О

Окунев А.С., 222
 Ольшанский М.Ю., 145
 Опарин Г.А., 80
 Орешкина Л.В., 242

П

Павский В.А., 152
 Павский К.В., 152
 Палюх Б.В., 109
 Пашинин А.А., 80
 Пелипец А.В., 55
 Пестов Д.А., 156
 Петров И.Б., 246
 Петровский А.И., 208
 Путилин А.Б., 250

С

Самодов И.О., 149
 Семерников Е.А., 38
 Семухина М.С., 109
 Серогодский А.И., 125
 Сесекин А.Н., 168
 Симонов А.С., 240

Соколов А.И., 136
 Соколова Ж.В., 250
 Степаненко С.А., 58
 Степанов М.В., 250
 Ступин Д.Д., 181
 Сузанский Д.Н., 160

Т

Талдыкин Е.В., 63, 67
 Таранов А.Ю., 145
 Тарасов И.Е., 70
 Телегин М.В., 63, 67
 Титов А.Г., 21
 Ткаченко И.В., 212
 Трегубов Д.Е., 162
 Тучин Д.А., 54

У

Угольников А.В., 76

Ф

Феокистов А.Г., 80
 Фитенко В.В., 54

Х

Хохлов Н.И., 246

Ч

Чекина М.Д., 258
 Чикин А.Л., 226
 Чикина Л.Г., 226

Ш

Шабас И.Н., 263
 Шелестенко Е.Ю., 18
 Шелестенко Е.Ю., 76
 Шишени А.В., 266
 Шляхов А.С., 168

Я

Якушенко Е.И., 212
 Ярошевский В.С., 54

СУПЕРКОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

СКТ-2014

Материалы 3-й Всероссийской
научно-технической конференции

29 сентября – 4 октября 2014
Дивноморское, Геленджик

ТОМ 2

Редактор В.О. Бронзов
Компьютерная верстка: Н.Ю. Иванова

Подписано в печать 5.08.2014.
Формат 60×84 ¹/₁₆. Бумага офсетная.
Печать офсетная. Усл. печ. л. 15,81. Уч.-изд. л.17,78.
Тираж 150 экз. Заказ 3737.

Издательство Южного федерального университета.

Отпечатано в отделе полиграфической, корпоративной и сувенирной продукции
Издательско-полиграфического комплекса КИБИ МЕДИА ЦЕНТРА ЮФУ
344090, г. Ростов-на-Дону, пр. Стачки, 200/1. Тел. (863) 247-80-51.



**Научно-исследовательский институт
многопроцессорных вычислительных систем
имени академика А.В. Каляева
федерального государственного автономного
учреждения высшего профессионального образования
“Южный федеральный университет”**

**Адрес: 347928, Россия, Ростовская область,
г. Таганрог ул. Чехова, 2, ГСП-284
тел./факс (863-4) 36-03-76, 615-459
e-mail: mvs@mvs.sfedu.ru
<http://www.mvs.sfedu.ru>**