ВЕСТНИК

ЮЖНО-УРАЛЬСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

2014
Т. 3, № 2

ISSN 2305-9052

СЕРИЯ

«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»

Учредитель — Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Южно-Уральский государственный университет» (национальный исследовательский университет)

Основной целью издания является пропаганда научных исследований в следующих областях:

- Вычислительная математика и численные методы
- Математическое программирование
- Распознавание образов
- Вычислительные методы линейной алгебры
- Решение обратных и некорректно поставленных задач
- Доказательные вычисления
- Численное решение дифференциальных и интегральных уравнений
- Исследование операций
- Теория игр
- Теория аппроксимации
- Информатика
- Математическое и программное обеспечение высокопроизводительных вычислительных систем
- Системное программирование
- Распределенные вычисления, облачные и грид-технологии
- Технология программирования
- Машинная графика
- Интернет-технологии
- Системы электронного обучения
- Технологии обработки баз данных и знаний
- Интеллектуальный анализ данных

Редакционная коллегия

Л.Б. Соколинский, д.ф.-м.н., проф., *отв. редактор*
В.П. Танана, д.ф.-м.н., проф., *зам. отв. редактора*
М.Л. Цымблер, к.ф.-м.н., доц., *отв. секретарь*
С.М. Абдуллаев, д.г.н., проф.
А.В. Панюков, д.ф.-м.н., проф.
К.С. Пан, *техн. секретарь*

Редакционный совет

В.И. Бердышев, д.ф.-м.н., акад. РАН, *председатель*
А. Андреяк, PhD, профессор (Германия)
В.В. Воеводин, д.ф.-м.н., чл.-кор. РАН
Дж. Донгарра, PhD, профессор (США)
И.И. Ерёмин, д.ф.-м.н., акад. РАН
А.Б. Куржанский, д.ф.-м.н., акад. РАН
В.Г. Романов, д.ф.-м.н., чл.-кор. РАН
Д. Маллманн, PhD, профессор (Германия)
А.Н. Томилин, д.ф.-м.н., профессор
В.Е. Третьяков, д.ф.-м.н., чл.-кор. РАН
А.М. Федотов, д.ф.-м.н., чл.-кор. РАН
В.И. Ухоботов, д.ф.-м.н., профессор
В.Н. Ушаков, д.ф.-м.н., чл.-кор. РАН
М.Ю. Хачай, д.ф.-м.н., профессор
П. Шумяцки, PhD, профессор (Бразилия)
Е. Ямазаки, PhD, профессор (Бразилия)

BULLETIN
OF THE SOUTH URAL STATE UNIVERSITY
SERIES "COMPUTATIONAL MATHEMATICS AND SOFTWARE ENGINEERING"

2014
Vol. 3, no. 2

ISSN 2305-9052

Vestnik Yuzhno-Ural'skogo Gosudarstvennogo Universiteta.
Seriya "Vychislitel'naya Matematika i Informatika"

South Ural State University

The main purpose of the series is publicity of scientific researches in the following areas:

- Numerical analysis and methods
- Mathematical optimization
- Pattern recognition
- Numerical methods of linear algebra
- Reverse and ill-posed problems solution
- Computer-assisted proofs
- Numerical solutions of differential and integral equations
- Operations research
- Game theory
- Approximation theory
- Computer science
- High performance computer software
- System programming
- Distributed, cloud and grid computing
- Programming technology
- Computer graphics
- Internet technologies
- E-learning
- Database and knowledge processing
- Data mining

Editorial Board

L.B. Sokolinsky, South Ural State University (Chelyabinsk, Russian Federation)
V.P. Tanana, South Ural State University (Chelyabinsk, Russian Federation)
M.L. Zymbler, South Ural State University (Chelyabinsk, Russian Federation)
S.M. Abdullaev, South Ural State University (Chelyabinsk, Russian Federation)
A.V. Panyukov, South Ural State University (Chelyabinsk, Russian Federation)
C.S. Pan, South Ural State University (Chelyabinsk, Russian Federation)

Editorial Counsil

V.I. Berdyshev, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russian Federation)
A. Andrzejak, Heidelberg University (Germany)
V.V. Voevodin, Lomonosov Moscow State University (Moscow, Russian Federation)
J. Dongarra, University of Tennessee (USA)
I.I. Eremin, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russian Federation)
A.B. Kurzhansky, Lomonosov Moscow State University (Moscow, Russian Federation)
V.G. Romanov, Sobolev Institute of Mathematics, Siberian Branch of the RAS (Novosibirsk, Russian Federation)
D. Mallmann, Julich Supercomputing Centre (Germany)
A.N. Tomilin, Institute for System Programming of the RAS (Moscow, Russian Federation)
V.E. Tretyakov, Ural Federal University (Yekaterinburg, Russian Federation)
A.M. Fedotov, Institute of Computational Technologies, SB RAS (Novosibirsk, Russian Federation)
V.I. Ukhobotov, Chelyabinsk State University (Chelyabinsk, Russian Federation)
V.N. Ushakov, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russian Federation)
M.Yu. Khachay, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russian Federation)
P. Shumyatsky, University of Brasilia (Brazil)
Y. Yamazaki, Federal University of Pelotas (Brazil)

Содержание

О НЕКОТОРЫХ ВАРИАНТАХ МЕТОДА ДЕКОМПОЗИЦИИ ОБЛАСТЕЙ
В.П. Ильин, Д.В. Перевозкин 5

ПРИМЕНЕНИЕ ГИБРИДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ К РЕШЕНИЮ ПРИКЛАДНЫХ ЗАДАЧ АЭРОГАЗОДИНАМИКИ
Е.Ю. Карцева, А.В. Кашковский, А.О. Лацис, А.В. Панасенко, Ю.П. Смольянов, В.В. Чернов, А.А. Шершнёв 20

МЕТОДЫ И СРЕДСТВА МЕТАМОНИТОРИНГА РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕД
И.А. Сидоров, А.П. Новопашин, Г.А. Опарин, В.В. Скоров 30

МЕТОДЫ И ЭВРИСТИКИ ПЛАНИРОВАНИЯ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЯХ С НЕОТЧУЖДАЕМЫМИ РЕСУРСАМИ
В.В. Топорков, А.В. Бобченков, Д.М. Емельянов, А.С. Целищев 43

ОЦЕНКА ПОПУЛЯРНОСТИ АВТОРОВ СОЦИАЛЬНОЙ СЕТИ С ПОМОЩЬЮ ПОИСКА ЭКСПЕРТОВ НА ПРИМЕРЕ СЕРВИСА TWITTER
Р.М. Миниахметов, Е.О. Цацина 63

ПРИМЕНЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СХЕМ ПОВЫШЕННОЙ ТОЧНОСТИ В ПРОЕКТИРОВАНИИ АНТЕННЫХ СИСТЕМ
А.Б. Хашимов, Р.Р. Салихов, Р.С. Альметов 77

Краткие сообщения

ИССЛЕДОВАНИЕ ГАЗОДИНАМИЧЕСКОГО ПОТОКА И КОНСТРУКЦИИ В МОДЕЛЬНОЙ ЭКСПЕРИМЕНТАЛЬНОЙ УСТАНОВКЕ
Л.Н. Бутымова, В.Я. Модорский 92

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ АЛГОРИТМ ШЕРМАНА – МОРРИСОНА ОБРАЩЕНИЯ МАТРИЦ НА GPU
Н.С. Недожогин, А.С. Сармакеева, С.П. Копысов 101

СОЗДАНИЕ СИМУЛЯТОРА ДЛЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ОБРАЗОВАТЕЛЬНОГО РЕСУРСНОГО ЦЕНТРА
П.Н. Полежаев, А.Л. Коннов, А.Е. Шухман 109

АВТОМАТИЧЕСКОЕ ОТОБРАЖЕНИЕ ПРОГРАММ НА ПРОЦЕССОР С ПЛИС-УСКОРИТЕЛЕМ
Д.В. Дубров, А.С. Рошаль, Б.Я. Штейнберг, Р.Б. Штейнберг 117

IMPACT OF WRF-3DVAR DATA ASSIMILATION ON THE PREDICTION OF RAINFALL OVER SOUTHERN BRAZIL
V.C. Beck, Y. Yamasaki, F.P. Härter 122

© Издательский центр ЮУрГУ, 2014

Contents

ON SOME VARIANTS OF DOMAIN DECOMPOSITION METHODS
V.P. Il'in, D.V. Perevozkin 5

APPLYING HETEROGENEOUS COMPUTING SYSTEM FOR SOLVING APPLIED PROBLEMS OF AEROGASDYNAMICS
E.Yu. Kartseva, A.V. Kashkovsky, A.O. Latsis, A.V. Panasenko, Yu.P. Smolyanov, V.V. Chernov, A.A. Shershnyov 20

METHODS AND TOOLS OF DISTRIBUTED COMPUTING SYSTEMS METAMONITORING
I.A. Sidorov, A.P. Novopashin, G.A. Oparin, V.V. Skorov 30

SCHEDULING METHODS AND HEURISTICS IN DISTRIBUTED COMPUTING WITH NON-DEDICATED RESOURCES
V.V. Toporkov, A.V. Bobchenkov, D.M. Yemelyanov, A.S. Tselishchev 43

TWITTER USERS POPULARITY ESTIMATION USING EXPERT FINDING
R.M. Miniakhmetov, E.O. Tsatsina 63

USING OF THE HIGH PRECISION NUMERICAL TECHNIQUES FOR ANTENNA SYSTEM DESIGN
A.B. Khashimov, R.R. Salikhov, R.S. Al'metov 77

Brief Reports

INVESTIGATION OF GAS DYNAMIC FLOW AND STRUCTURE IN EXPERIMENTAL EQUIPMENT
L.N. Butymova, V.Ya. Modorsky 92

SHERMAN – MORRISON HIGH-PERFORMANCE ALGORITHM FOR INVERSE MATRIX ON GPU
N.S. Nedozogin, A.S. Sarmakeeva, S.P. Kopysov 101

DEVELOPING SIMULATOR OF EDUCATIONAL RESOURCE DATACENTER
P.N. Polezhaev, A.L. Konnov, A.E. Shukhman 109

AUTOMATIC MAPPING PROGRAMS ONTO A PROCESSOR WITH AN FPGA ACCELERATOR
D.V. Dubrov, A.S. Roshal, B.Ya. Steinberg, R.B. Steinberg 117

IMPACT OF WRF-3DVAR DATA ASSIMILATION ON THE PREDICTION OF RAINFALL OVER SOUTHERN BRAZIL
V.C. Beck, Y. Yamasaki, F.P. Härter 122

УДК 519.612

О НЕКОТОРЫХ ВАРИАНТАХ МЕТОДА ДЕКОМПОЗИЦИИ ОБЛАСТЕЙ[1]

В.П. Ильин, Д.В. Перевозкин

Рассматриваются алгоритмы масштабируемого распараллеливания решения сверхбольших разреженных сеточных СЛАУ, представленных в универсальных сжатых форматах, в том смысле, что их реализация осуществляется без программных ограничений на порядки алгебраических систем и на количество используемых вычислительных узлов, процессоров и/или ядер.

Данная задача сводится к распределенному варианту алгебраической 3D-декомпозиции областей, в котором отсутствует чрезмерная расчетно-информационная нагрузка корневого процессора, т.е. все организуемые MPI-процессы, каждый из которых соответствует своей подобласти, являются практически равноправными. Вычислительный процесс состоит из двух основных этапов, первый из которых заключается в непосредственной автоматической декомпозиции, на основе анализа матричного портрета и формировании крупноблочного представления СЛАУ. Второй этап — это реализация крыловского итерационного алгоритма FGMRES (гибкого обобщенного метода минимальных невязок), использующего точное или приближенное обращение диагональных матричных блоков (многопоточное решение подсистем в подобластях с использованием средств OpenMP) с помощью прямого или итерационного метода соответственно.

Описываемые методы реализованы в составе библиотеки алгебраических решателей Krylov. В работе приводятся некоторые оценки используемых ресурсов и особенности параллельных вычислительных технологий. Эффективность разработанных алгоритмов иллюстрируется результатами численных экспериментов по решению характерных алгебраических задач на различных конфигурациях многопроцессорной вычислительной системы.

Ключевые слова: декомпозиция областей, матричный граф, распараллеливание алгоритмов, сеточные уравнения, разреженные СЛАУ, программные и вычислительные технологии

Введение

Методы декомпозиции областей (МДО), как главное орудие масштабируемого распараллеливания численного решения больших задач математического моделирования на многопроцессорных вычислительных системах (МВС), является предметом изучения огромного количества статей и монографий, представленных, например, на специальном интернет-сайте [1].

Обширную область изучаемых здесь вопросов можно разбить на две основные части. Первая из них относится к построению новых алгоритмов для различных классов задач, их теоретическому обоснованию, оценкам скорости сходимости, оптимизации и сравнительному анализу. Относительно свежие математические результаты в данных направлениях можно найти на сайте 22-й Международной конференции по декомпозиции областей [2], состоявшейся в сентябре 2013 г.

Вторая часть не менее актуальных проблем — это эффективное отображение МДО на архитектуру современных МВС, которая, несмотря на стремительное обновление суперкомпьютерного списка TOP-500, в ближайшую пятилетку, до рождения первого экзафлопсни-

[1] Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии 2014»

ка, прогнозируется традиционной: кластер из гетерогенных узлов, содержащих по несколько центральных многоядерных процессоров (CPU) с общей памятью, а также по несколько графических процессоров общего назначения (GPGPU), сверхбыстрых, но с достаточно сложным использованием. Управление иерархической памятью с распределенным и общим доступом ограничивается средствами MPI, OpenMP и CUDA.

Цель данной работы заключается в том, чтобы хотя для достаточно ограниченного набора алгоритмов декомпозиции сконцентрироваться на вариантах технологических решений, обеспечивающих реализацию в определенном смысле главной задачи — создания высокопроизводительного математического и программного обеспечения, не содержащего формальных ограничений на число степеней свободы и на количество используемых вычислительных ядер и/или процессоров.

Рассматриваемая нами задача с функциональной точки зрения достаточно простая — это численное решение систем линейных алгебраических уравнений (СЛАУ) с разреженными матрицами больших порядков (до 10^8–10^{11}), возникающими при различных аппроксимациях многомерных междисциплинарных начально-краевых задач со сложной геометрической конфигурацией границ на неструктурированных сетках.

Особенность проблемы заключается в объеме данных, необходимых для хранения ненулевых элементов разреженных матриц, число которых в каждой из строк не зависит от размерности СЛАУ и составляет обычно несколько десятков (их количество растет с увеличением порядка точности аппроксимации). Например, система уравнений Навье —Стокса, описывающая гидро-газодинамические течения, включает 4 неизвестные скалярные функции — три компоненты вектора скорости и давление. Поэтому на трехмерной сетке с числом узлов $1000^3 = 10^9$ общий порядок СЛАУ составляет примерно $4 \cdot 10^9$, а количество ненулевых матричных элементов — около 10^{11}. Поскольку возникающие при этом вычислительные задачи являются очень плохо обусловленными, то расчеты необходимо проводить, как минимум, со стандартной двойной точностью (64 двоичных разряда, или 8 байтов, на одно число). Таким образом, размещение соответствующей СЛАУ на одном процессоре, объем оперативной памяти которого составляет обычно несколько гигабайт, становится проблематичным. В этом случае не проходит зачастую применяемая стратегия распараллеливания алгебраических задач, когда сначала глобальная матрица формируется на одном (головном) процессоре, а потом ее блоки рассылаются по остальным используемым процессам. Единственная альтернатива здесь — распределенное хранение матрицы, когда ее элементы изначально рассчитываются в разных процессорах.

Данная статья построена следующим образом. В разделе 1 мы представляем основные структурные особенности рассматриваемых СЛАУ и применяемых МДО. В разделе 2 описываются некоторые предлагаемые параллельные алгоритмы и их программные реализации. В разделе 3 приводятся результаты характерных численных экспериментов по решению представительной серии методических задач.

1. Некоторые постановки, формализмы и алгоритмы декомпозиции

Рассматривается следующая задача: пусть матрица СЛАУ

$$Au = f, \;\; A = \{a_{i,j}\} \in \mathcal{R}^{N,N}, \tag{1}$$

разбита на блочные строки

$$A = \{A_p \in \mathcal{R}^{N_p,N},\ p = 1, ..., P\},\ \ N_1 + ... + N_p = N, \tag{2}$$

с приблизительно равным числом строк $N_p \gg 1$ в каждой. Соответствующим образом также разбиваются векторы искомого решения и правой части

$$u = \{u_p \in \mathcal{R}^{N_p}\},\ \ f = \{f_p \in \mathcal{R}^{N_p}\}, \tag{3}$$

так что система уравнений (1) может быть записана в форме совокупности P подсистем

$$A_{p,p}u_p + \sum_{\substack{q=1 \\ q\neq p}}^{P} A_{p,q}u_q = f_p,\ \ p = 1, ..., P, \tag{4}$$

где $A_{p,q} \in \mathcal{R}^{N_p,N_q}$ — прямоугольные матричные блоки, получаемые при разбиении каждой блочной строки (и всей матрицы A) на блочные столбцы:

$$A_p = \{A_{p,q}; q = 1, ..., P\},\ \ A = \{A_{p,q}; p, q = 1, ..., P\}. \tag{5}$$

Будем считать, что СЛАУ (1) представляет собой систему сеточных уравнений, аппроксимирующих некоторую многомерную краевую задачу для дифференциального уравнения, так что каждая компонента векторов u, f соответствует узлу сетки, общее число которых в расчетной сеточной области $\Omega^h = \bigcup\limits_{p=1}^{P} \Omega_p$ равно N. Отметим, что в методах конечных элементов (МКЭ, [4]) такие системы называются узловыми, но более сложные сеточные уравнения мы пока не рассматриваем. При этом блочное разбиение матриц и векторов соответствует разбиению (декомпозиции) Ω^h на P сеточных непересекающихся подобластей Ω_p, в каждой из которых находится N_p узлов, $N_1 + ... + N_P = N$.

Описанная декомпозиция Ω^h не использует узлов-разделителей, т.е. условные границы подобластей не проходят через узлы сетки. Формальности ради можно считать, что внешность расчетной области есть неограниченная подобласть Ω_0 с числом узлов $N_0 = 0$.

Сеточное уравнение для i-го узла сетки может быть записано в виде

$$a_{i,i}u_i + \sum_{j\in\omega_i,\ j\neq i} a_{i,j}u_j = f_i,\ \ i \in \Omega^h, \tag{6}$$

где через ω_i обозначается сеточный шаблон i-го узла, т.е. совокупность номеров всех узлов (как правило, являющихся геометрическими соседями), участвующих в i-м уравнении.

Если сеточные узлы и соответствующие переменные пронумерованы следующим образом: сначала идут подряд все узлы 1-й подобласти Ω_1 (неважно в каком внутреннем порядке), затем — узлы второй подобласти и т.д., — то вышеприведенное блочное разбиение матриц и векторов, приводящее к подсистемам (4), соответствует декомпозиции расчетной области без пересечения подобластей. При этом номера узлов из ω_i, соседних к i-му узлу, указываются в CSR-формате [3] для i-й строки матрицы (фактически там указываются номера столбцов как для диагонального, так и внедиагональных элементов). Отметим, что взаимнооднозначного соответствия алгебраической и геометрической (сеточной) интерпретации структуры СЛАУ может и не существовать. Типичный пример — одному узлу сетки может соответствовать $m > 1$ переменных в алгебраической системе. Соответствующая ситуация имеет место в междисциплинарных задачах, описываемых системами из m дифференциальных уравнений, у которых решение есть вектор-функция размерности m.

При заданном блочном разбиении СЛАУ, или декомпозиции сеточной расчетной области без пересечения подобластей, для решения системы (1) можно построить хорошо распараллеливаемый аддитивный метод Шварца, представимый итерационным процессом

$$A_{p,p}u_p^n = f_p - \sum_{\substack{q=1 \\ q\neq p}}^{P} A_{p,q}u_q^{n-1} \equiv g_p^{n-1}. \tag{7}$$

При вычисленных правых частях g_p^{n-1} нахождение всех u_p^n требует для каждой подобласти решения подсистем с порядками N_p, которые на каждой n-й итерации могут выполняться независимо и одновременно на разных процессорах. В данном случае перевычисление g_p^{n-1} фактически означает использование новых значений u_p^{n-1} для соседних подобластей в качестве граничного условия 1-го рода (Дирихле) для околограничных внутренних узлов из Ω_p.

Рассмотренное двойственное представление структуры СЛАУ (1) может быть дополнено путем введения графового описания этой же задачи. Каждому сеточному узлу с номером i (или i-й строке матрицы A) можно сопоставить вершину v_i некоторого графа G, а матричному элементу $a_{i,j}$ — ребро графа $G = (V, E)$ и в соответствии с [5] определить множества вершин и ребер:

$$V = \{v_i;\ i = 1, ..., N\},\ \ E = \{(v_i, v_j)|a_{i,j} \neq 0\}. \tag{8}$$

Отметим следующий существенный технологический аспект, возникающих при реализации численного решения подсистемы (7). Традиционно универсальные программные решатели СЛАУ, ориентированные на матричный CSR-формат, предполагают сплошную нумерацию векторных компонент и соответствующих матричных строк, начиная с единицы или нуля. Однако, если для исходной СЛАУ информация дана в глобальной нумерации переменных, то в подобласти Ω_p требуется своя локальная нумерация. Таким образом, здесь возникает техническая задача переформирования CSR_p-формата для диагонального блока $A_{p,p}$ с переходом из глобальной нумерации в локальную. А после вычисления подвекторов u_p в подобластях, естественно, потребуется сборка глобального вектора u, т.е. перевод векторных компонент из локальной нумерации в глобальную.

Известно, что скорость сходимости итераций метода Шварца (7) возрастет, если декомпозицию расчетной области сделать с пересечением подобластей.

В силу этого мы дадим определение расширенной сеточной подобласти $\bar{\Omega}_p \supset \Omega_p$ с пересечениями, величину которых мы будем определять в терминах количества околограничных сеточных слоев, или фронтов.

Обозначим через $\Gamma_p^0 \in \Omega_p$ множество внутренних околограничных узлов из Ω_p, т.е. таких узлов $P_i \in \Omega_p$, у которых один из соседей не лежит в $\Omega_p (P_j \notin \Omega_p,\ j \in \omega_i,\ j \neq i))$. В Γ_p^0 определим подмножество узлов $\Gamma_{p,q}^0$, у которых соседние по шаблону узлы принадлежат смежной подобласти $\Omega_q, q \in Q_p^0$, где Q_p^0 — совокупность номеров подобластей, соседних с Ω_p. Таким образом,

$$\Gamma_p^0 = \bigcup_{q\in Q_p^0} \Gamma_{p,q}^0, \tag{9}$$

причем подмножества $\Gamma_{p,q}^0$ могут пересекаться, т.е. содержать околограничные узлы, у которых есть соседи из различных смежных подобластей.

Обозначим далее через Γ_p^1 множество узлов, соседних с узлами из Γ_p^0, но не принадлежащих Ω_p, через Γ_p^2 — множество узлов, соседних с узлами из Γ_p^1, но не принадлежащих объединению $\Gamma_p^1 \bigcup \Omega_p$, через Γ_p^3 — множество узлов, соседних с Γ_p^2, но не принадлежащих

$\Gamma_p^2 \bigcup \Gamma_p^1 \bigcup \Omega_p$, и т.д. Соответственно эти множества назовем первым внешним слоем (фронтом) узлов, вторым слоем, третьим и т.д. Получаемое объединение узлов

$$\Omega_p^\Delta = \Omega_p \bigcup \Gamma_p^1 ... \bigcup \Gamma_p^\Delta \tag{10}$$

будем называть расширенной p-й сеточной подобластью, а целую величину Δ определяем как величину расширения, или пересечения (в терминах количества сеточных слоев). Случай $\Delta = 0$ фактически означает декомпозицию области Ω^h на подобласти без пересечений ($\Omega_p^0 = \Omega_p$).

Множество $\Gamma_p^\Delta \in \Omega_p^\Delta$ представляет совокупность внутренних околограничных узлов расширенной подобласти Ω_p^Δ, а $\Gamma_p^{\Delta+1}$ есть множество внешних околограничных узлов. Условная граница Ω_p^Δ проходит между Γ_p^Δ и $\Gamma_p^{\Delta+1}$.

Аналогично Γ_p^0, множество Γ_p^Δ можно разбить на подмножества околограничных узлов

$$\Gamma_p^\Delta = \Gamma_{p,q_1}^\Delta \bigcup \Gamma_{p,q_2}^\Delta ... \bigcup \Gamma_{p,q_{m_p}}^\Delta \tag{11}$$

у которых соседние (по шаблону) находятся соответственно в подобластях $\Omega_{q_1}, \Omega_{q_2}, ..., \Omega_{q_{m_p}}$ (здесь m_p означает количество подобластей, пересекающихся с Ω_p^Δ, а $q_1, q_2, ..., q_{m_p}$ — номера этих подобластей). Отметим, что Γ_p^Δ может содержать «кратные» околограничные узлы, принадлежащие одновременно разным подмножествам $\Gamma_{p,q}^\Delta$, т.е. у которых есть соседи из различных смежных подобластей.

Если обозначить через $\Omega_{p,q}^\Delta$ пересечение какой-то подобласти Ω_q с расширенной подобластью $\bar{\Omega}_p$, то последнюю можно представить в форме объединения

$$\bar{\Omega}_p = \Omega_p \bigcup \Omega_{p,q_1}^\Delta \cdots \bigcup \Omega_{p,q_{m_1}}^\Delta, \tag{12}$$

причем каждое пересечение составляется из частей внешних околограничных фронтов:

$$\Omega_{p,q}^\Delta = \Gamma_{p,q}^1 \bigcup \Gamma_{p,q}^2 ... \bigcup \Gamma_{p,q}^\Delta, \;\; q = q_1, ..., q_{m_k}. \tag{13}$$

Таким образом, сборка матрицы СЛАУ для расширенной подобласти может быть интерпретирована как идентификация, объединение и перенумерация множества узлов, участвующих в (12), (13).

На формальном алгебраическом языке расширенные подсистемы уравнений и итерационный метод Шварца записываются аналогично (7):

$$\bar{A}_{p,p} \bar{u}_p^n = \bar{f}_p - \sum_{\substack{q=1 \\ q \neq p}}^{P} \bar{A}_{p,q} \bar{u}_q^{n-1} = \bar{g}_p^{n-1}. \tag{14}$$

Здесь размерность векторов $\bar{u}_p^n$ и $\bar{f}_p$ равна числу узлов $\bar{N}_p$ в расширенной подобласти $\bar{\Omega}_p$. При этом надо иметь в виду, что в пересечениях $\Omega_{p,q}^\Delta$ один узел принадлежит двум или более подобластям.

Остановимся теперь подробнее на формировании правых частей $\bar{g}_p^{n-1}$ или g_p^{n-1} в (7), (13). Для этого рассмотрим сеточное уравнение вида (4) для расположенного в $\bar{\Omega}_p$ i-го околограничного узла, у которого некоторые из соседей находятся в других подобластях $\Omega_q, q \neq p$:

$$a_{i,i} u_i + \sum_{j \in \Omega_p} a_{i,j} u_j = f_i - \sum_{j \notin \Omega_p} a_{i,j} u_j. \tag{15}$$

Отметим, что в (15) i-й узел располагается и в $\bar{\Omega}_p$, и в какой-то из соседних подобластей. Отсюда строим итерационный процесс, несколько модифицировав данное уравнение:

$$\Big(a_{i,i} + \theta \sum_{j \notin \Omega_p} a_{i,j}\Big) u_i^n + \sum_{j \in \Omega_p} a_{i,j} u_j^n = f_i + \sum_{j \notin \Omega_p} a_{i,j} (\theta u_i^{n-1} - u_j^{n-1}). \tag{16}$$

Здесь $\theta \in [0,1]$ — некоторый итерационный параметр, который при $\theta = 0$ соответствует условию Дирихле в околограничном i-м узле, при $\theta = 1$ — условию Неймана, а при $0 < \theta < 1$ — условию 3-го рода, или Робена. Значения u_i^{n-1}, u_j^{n-1} в правой части (16) берутся с предыдущей итерации в узлах из соседних к Ω_p подобластей. Очевидно, что если итерационный процесс (16) сходится, то при любом θ его предел будет решением исходной системы (15).

Если ввести диагональную $D_p = diag\{\sum\limits_{j \neq \Omega_p} a_{i,j}\}$, то итерационный процесс (16) можно записать в форме, обобщающей соотношение (14):

$$\bar{B}_p(\bar{u}_p^n - \bar{u}_p^{n-1}) = \bar{r}_p^{n-1} \equiv \bar{f}_p - (A\bar{u}^{n-1})_p, \tag{17}$$

где предобуславливающая матрица $\bar{B}_p$ имеет вид $\bar{B}_p = \bar{A}_{p,p} + \theta D_p$.

В приведенных выше рассуждениях мы рассмотрели расширение подобласти Ω_p в наружную сторону. Однако такие же процедуры осуществляются и для соседних подобластей, что приводит к построению пересечений внутри самой Ω_p. Их реализация может осуществляться по сеточных слоям (фронтам), как и при расширении Ω_p.

Остановимся коротко на вопросах декомпозиции при решении нестационарных задач, когда использование неявных аппроксимаций по времени приводит к необходимости решения СЛАУ на каждом шаге. В этом случае приходится решать последовательно алгебраические системы с разными правыми частями и одинаковыми матрицами, если их элементы не меняются со временем.

2. Особенности реализации распределенных алгоритмов

В высокопроизводительных версиях декомпозиции областей существуют два ключевых момента: формирование разреженных алгебраических систем для соответствующих подобластей и реализация параллельных по подобластям крыловских итерационных процессов якобиевого типа.

2.1. Метод построения квазисбалансированных сеточных подобластей

В начале данного раздела остановимся на одном факторе, имеющем большое значение для скорости сходимости итерационных процессов вида (7) по подобластям. Речь идет о топологии и некоторых других характеристиках разбиения расчетной сеточной области Ω^h на сеточные подобласти Ω_p^h, $p = 1, ..., P$. Здесь следует отметить, что вообще говоря, нужно различать исходную непрерывную расчетную область Ω, в которой решаются дифференциальные и/или интегральные уравнения, и ее дискретный аналог Ω^h. Более того, декомпозицию можно делать в первую очередь: сначала разбить область на подобласти $\Big(\Omega = \bigcup\limits_{p=1}^{P} \Omega_p\Big)$, а затем в каждой из них строить подсетку Ω_p^h, которая геометрически привязана к соответствующей Ω_p. Однако мы эти аспекты сознательно опускаем, ограничиваясь только декомпозицией сеточной области.

Совокупность подобластей образует собой макросеть, или макрограф, у которого каждая подобласть представляет собой макровершину, а связи с геометрически соседними подобластями — макроребра. Как известно [6], скорость сходимости итераций по подобластям будет тем выше, чем меньше диаметр макрографа (максимальное из минимально возможных расстояний между макроузлами). При разбиении трехмерной области на достаточно большое число P подобластей диаметр макрографа есть $O(P)$ при одномерной декомпозиции, $O(P^{1/2})$ — при двумерной и $O(P^{1/3})$ — при трехмерной. Поэтому естественно, что оптимальной является трехмерная сбалансированная декомпозиция, при которой число узлов в каждой подобласти является примерно одинаковым, чтобы при одновременном решении алгебраических подсистем в подобластях соответствующие процессоры имели минимальные простои.

В некотором смысле альтернативным подходом к декомпозиции сеточной области является следующий. В соответствии с вышеприведенными рассуждениями, оптимальное разбиение кубической области с кубической же сеткой — это декомпозиция на кубические сеточные подобласти. Особенность этой конфигурации заключается в том, что при заданном числе узлов она имеет относительно малый диаметр соответствующего сеточного подграфа, т.е. сравнительно близкий к идеальному варианту — сфере. Именно это наблюдение мы и положим в основу рассматриваемого «локального» принципа декомпозиции: строить сеточные подобласти с примерно равным количеством узлов и имеющие относительно малые диаметры своих сеточных подграфов.

Итак, описываемый алгоритм декомпозиции преследует цель создания сеточных подобластей Ω_p, $p = 1, ..., P$, по возможности меньшего диаметра, где диаметр понимается как диаметр подграфа $G_p(V_p, E_p)$, порожденного [7] множествами содержащихся в подобласти вершин V_p и ребер E_p. Вообще говоря, поиск оптимального в терминах какого-либо критерия разбиения графа G является задачей как минимум полиномиальной сложности, хотя для решения больших СЛАУ нежелательна даже квадратичная зависимость от количества элементов графа N^V, N^E.

С этой точки зрения интересен следующий ход: с помощью какого-либо экономичного метода выполнить на первом этапе разбиение графа на связные непересекающиеся подмножества сравнительно небольшой мощности и малого диаметра, впоследствии рассматривая их как макровершины и декларируя наличие ребра между двумя соседними макровершинами в том и только том случае, если есть хотя бы одно ребро, соединяющее некоторые элементы этих двух подмножеств. В дальнейшем предлагается интерпретировать полученный таким образом граф как огрубленную (агрегированную) копию исходного, применяя более сложные (или, наоборот, простые) алгоритмы для формирования последующих и конечного разбиений.

Не останавливаясь пока на методах агрегации каждого этапа, отметим, что предлагаемый подход является многошаговым и фактически образует последовательность вложенных сеток, или макросеток, из подобластей. Если обозначить через s номер шага в рассматриваемой иерархической структуре, то для формализации данного процесса можно ввести следующие обозначения: $G^{(s)}(V^{(s)}, E^{(s)})$, $s = 0, 1, ..., M$, — граф макросетки s-го уровня ($s = 0$ соответствует начальному сеточному графу, в котором вершина представляет простой узел сетки, а M есть количество стадий агрегации, которое заранее неизвестно), $N_V^{(s)}$ и $N_E^{(s)}$ — число макровершин и макроребер макросетки s-го уровня.

В целях построения начального разбиения предлагается использовать поиск в ширину [5]. Данный метод обладает привлекательным свойством помечать в первую очередь вершины с наименьшим расстоянием от исходной, что соответствует цели создания подмножеств малого диаметра. Поиск ведется среди вершин, еще не относящихся к какому-либо подмножеству, до тех пор, пока количество помеченных вершин не достигнет определенного предела (желаемого размера макровершины $N_{\max}^{(s)}$, где $N_{\max}^{(0)}$ означает число узлов сетки, которое допускается включать в макровершину первого уровня). Все помеченные на j-м шаге алгоритма вершины объявляются принадлежащими j-му подмножеству, после чего процедура повторяется до исчерпания множества вершин.

Формализуем алгоритм поиска в ширину. Через $Adj(v)$ обозначим множество вершин, смежных вершине v. Пусть также Q — структура данных типа очередь [5]. Обозначим через $Q \leftarrow S$ добавление множества S к очереди (в произвольном порядке), $v \leftarrow Q$ — изъятие элемента из очереди. Для выполнения поиска в ширину дополнительно введем C и W — целочисленные массивы (списки) цветов и весов вершин (смысл последних будет раскрыт позднее). Через $C(v)$ и $W(v)$ обозначим текущие цвет и вес вершины v. Перед стартом алгоритма цвет всех вершин инициализируется нулем, а их вес полагаем равным единице. Также введем $n_{\max}$ — желаемый размер макровершины. Тогда алгоритм заключается в начальном выборе произвольной вершины v из множества V и в дальнейших преобразованиях графа в соответствии с псевдокодом, представленным на рис.

$i = 1$
while $\{u \in V \mid C(u) = 0\} \neq \emptyset$
  pick any v *from* $\{u \in V \mid C(u) = 0\}$
  $Q := v$
  $n = 0$
  while $(n < n_{max}$ *and* $Q \neq \emptyset)$
    $v \leftarrow Q$
    $C(v) := i$
    $Q \leftarrow Adj(v) \cap \{u \in V \mid C(u) = 0\}$
    $n = n + W(v)$
  end while
  $i = i + 1$
end while

После работы алгоритма объявим $V_i = \{u \mid C(u) = i\}$. Сформируем граф с макровершинами $G' = (V', E')$, где $V' = \{V_1, \ldots V_{N'}\}$, N' — количество получившихся макровершин, а множество ребер E' определяется так:

$$E' = \{(V_i, V_j) \mid \exists u \in V_i, v \in V_j \mid (u, v) \in E\}. \tag{18}$$

Указанный выше алгоритм можно использовать и для формирования собственно подобластей нужного размера, положив $W(V_i) = |V_i|$, а $n_{\max}$ равным желаемому размеру подобласти. На данный момент именно такой алгоритм используется в библиотеке Krylov, хотя, вообще говоря, представляется целесообразным использование некоторого аналога очереди с приоритетом, где, например, приоритет тем выше, чем больше ребер связывают рассматриваемую макровершину с текущим составом формируемой подобласти, и тем ниже,

чем меньше вес (мощность) этой макровершины (стоит заметить, что в таком случае разумно вычислять приоритет макровершин в момент извлечения элемента из очереди, а не в момент добавления). Такой алгоритм будет отдавать предпочтение макровершинам с размером меньше типичного, выполняя их поглощение, и тем макровершинам, включение которых способствует уменьшению «площади» границы подобласти (т.е. количества ребер, связывающих эту подобласть с остальными).

Предполагается, что созданные таким образом подобласти будут обеспечивать хорошую скорость сходимости метода Шварца.

2.2. Особенности программной реализации

Рассмотрим более подробно некоторые вопросы, касающиеся непосредственно организации вычислительного процесса, причем остановимся для краткости на алгоритмах без пересечения подобластей.

В первую очередь заметим, что выполнение одного шага аддитивного метода Шварца (14) состоит из P независимых операций решения СЛАУ, для реализации каждой из которых формируется MPI-процесс, а выполнение ее осуществляется одним из процессоров CPU с указываемым количеством доступных вычислительных ядер, или потоков N_{th}, которыми выполняется внутреннее распараллеливание над общей памятью средствами OpenMP.

Во вторых, реально итерации по подобластям проводятся не по алгоритму Шварца (14), а с помощью предобусловленного гибкого обобщенного метода минимальных невязок FGMRES [8], в котором блочно-диагональная матрица с блоками $A_{p,p}$ на главной диагонали выступает в качестве предобуславливателя. Решение подсистемы для каждой p-й подобласти осуществляется прямым алгоритмом или итерационным, и в последнем случае предобуславливатель фактически является переменным.

Учитывая все вышесказанное, логичной представляется следующая схема реализации вычислений. Будем хранить векторы так же, как и матрицу, по N_p компонент в каждом процессе, соответствующем своей подобласти. В таком случае векторно-векторные операции и обращение предобуславливающего оператора можно выполнять без обмена данными между MPI-процессами (исключение составляет лишь скалярное произведение, где объем пересылаемых данных невелик). Для вычисления действия оператора A на вектор будем пересылать лишь те данные, которые требуются для вычисления соответствующих частей вектора, т.е. только между соседними подобластями.

Итерации по подобластям продолжаются до выполнения условий

$$||r^n|| = ||f - Au^n|| \le \varepsilon^e ||f||, \quad \varepsilon^e \ll 1, \quad \text{или} \quad n \le n^e_{\max}, \tag{19}$$

где ε^e и $n^e_{\max}$ — задаваемые критерии останова. Если решение вспомогательных СЛАУ в подобластях осуществляется итерационным методом, то для него задаются внутренние критерии ε^i, $n^i_{\max}$.

3. Примеры численных экспериментов

Главная задача данного раздела заключается в том, чтобы продемонстрировать эффективность итерационного метода по подобластям, которая в значительной степени зависит от качества декомпозиции. Расчеты проводились на кластере ИВМиМГ СО РАН [9] для

количества подобластей $P = 1, 2, 4, 8, 16, 32$, причем случай $P = 1$ соответствует решению задачи без применения декомпозиции.

В табл. 1 мы приводим результаты по решению модельной задачи Дирихле для диффузионно-конвективного уравнения в декартовой системе координат x, y, z с постоянными конвективными коэффициентами $p = q = r = 16$, см. [10, 11], в кубической области $\Omega = [0, 1]^3$, аппроксимируемой на кубической сетке с числом узлов $N = N_x^3$ с помощью экспоненциальной семиточечной схемы. Граничные условия соответствуют точному решению $u(x, y, z) = 1$, а начальное приближение выбрано $u^0(x, y, z) = 0$. Для критерия останова (19) внешнего итерационного процесса во всех случаях было выбрано $\varepsilon^e = 10^{-7}$.

Отметим, что в приведенных расчетах использовался двухуровневый вариант декомпозиции ($s = 0, 1$ в разделе 2), причем выбирались значения $N_{\max}^{(0)} = \sqrt{N}$, $N_{\max}^{(1)} = N/P$.

В каждой клетке табл. 1 представлены сверху вниз: количество внешних итераций, астрономическое и процессорное время (t_a и t_{cpu}, последнее — суммарное по всем MPI-процессам и по всем вычислительным потокам). В качестве решателя в подобластях использовался PARDISO из библиотеки Intel MKL [3], при заданном числе потоков $N_{th} = 12$. Из-за высоких требований данного решателя к объему оперативной памяти оказалось невозможным получить на имеющемся оборудовании результаты, соответствующие незаполненным клеткам таблицы.

Таблица 1

Решение модельной задачи с использованием PARDISO в подобластях

$N \setminus P$	1	2	4	8	16	32
64^3	1	29	32	39	54	78
	32,2	14,2	6,39	2,88	3,42	1,58
	164	93,5	55,1	46,3	232	200
128^3	1	40	44	53	75	108
	885	221	86,6	30,1	20,4	12,6
	10340	4987	1785	782	1232	1615
256^3	—	—	60	72	102	142
			2222	332	212	138
			100009	13450	18400	18210

В табл. 2 приводятся результаты для решения той же задачи, однако для решения СЛАУ в подобластях использовался предобусловленный метод бисопряженных градиентов со стабилизацией (BiCGStab), причем для предобуславливания использовалась модификация Айзенштата [4, 8] алгоритма неполной факторизации, без использования какого-либо распараллеливания. Критерии останова полагались следующими: $\varepsilon^i = 10^{-1}$, $n_{\max}^i = 10$.

В табл. 3 представлена зависимость между критериями останова ε^i, $n_{\max}^i$, фактически отвечающих за точность решения СЛАУ в подобластях, и величинами, отражающими скорость сходимости внешнего итерационного процесса, а также общую вычислительную сложность. В каждой клетке таблицы перечислены: количество внешних итераций n^e, общее количество внутренних итераций n^i (суммарное по всем подобластям и по всем внешним итерациям), астрономическое и процессорное время в секундах. Для сравнения также

Таблица 2

Решение модельной задачи с использованием предобусловленного итерационного метода BiCGStab в подобластях, $\varepsilon^i = 0,1$, $n^i_{\max} = 10$

$N \setminus P$	1	2	4	8	16	32
64^3	9	29	34	47	65	98
	4,05	6,35	3,44	1,54	1,59	2,26
	4,05	12,7	13,2	12,3	20,9	77,3
128^3	18	58	56	67	92	132
	64,9	115	57,4	25,2	15,9	18,3
	64,9	230	229	201	215	510
256^3	21	64	74	92	143	92
	606	1045	647	443	271	231
	606	2079	2588	3553	3916	6627

приводятся величины, получаемые при решении СЛАУ в подобластях прямым методом PARDISO.

Таблица 3

Влияние точности решения СЛАУ в подобластях на скорость сходимости и ресурсоемкость

Задача \ критерии	$\varepsilon^i = 0,1$ $n^i_{\max} = 10$	$\varepsilon^i = 0,01$ $n^i_{\max} = 20$	$\varepsilon^i = 0,001$ $n^i_{\max} = 40$	PARDISO
$P = 8$ $N = 128^3$	67	57	54	53
	3297	5280	7040	424
	25,2	39,9	52,2	30,1
	201	318	415	782
$P = 16$ $N = 256^3$	143	111	108	102
	15929	18843	25489	1632
	271	319	427	212
	3916	5112	6824	18400

Отметим, что во всех приведенных экспериментах итоговая погрешность численного решения была примерно одинаковой и равнялась 10^{-6}, что достаточно хорошо соответствует критерию $\varepsilon^e = 10^{-7}$.

По приведенным результатам можно сделать следующие выводы.

- С увеличением количества подобластей P и соотвествующих им MPI-процессов, до определенного числа наблюдается существенное ускорение решения СЛАУ как для прямого, так и итерационного алгоритмов в подобластях. Однако, в силу роста числа внешних итераций n^e с увеличением P, а также ввиду роста количества обменов при этом, наблюдается минимум $t_a(P)$, который на грубой сетке с $N = 64^3$ наступает при $P = 8$, а с увеличением N проявляется при гораздо большем числе P. Уменьшения числа n^e можно достичь за счет применения декомпозиции с пересечением подобластей и использования других приемов ускорения итераций по подобластям, однако

вопрос о возможности монотонного роста ускорения вычислений с ростом P остается открытым.

- Сравнение времен решения СЛАУ при использовании для решения СЛАУ в подобластях прямого или итерационного алгоритмов показывает, что в различных ситуациях выигрывает то один, то другой подход. Как и следовало ожидать из теоретических оценок вычислительной сложности, прямой алгоритм проигрывает итерационному, если число узлов в подобласти достаточно велико. Однако с ростом значений P ресурсоемкость прямого метода убывает быстрее, чем итерационного. Справедливости ради следует заметить, что времена счета соответствуют реализации прямого алгоритма с распараллеливанием по 12 потокам, а итерационного — без распараллеливания.
- Что касается решения СЛАУ с применением итерационного алгоритма в подобластях, то здесь существенные резервы ускорения вычислений заложены в оптимизации критериев окончания внутренних итераций. Как видно из табл. 3, уменьшение ε^i и увеличение $n^i_{\max}$ приводит к излишним итерациям в подобластях и увеличению времени счета. Приведенные результаты носят предварительный характер и оставляют открытым вопрос об оптимальной стратегии выбора критериев останова внутренних итераций.

Заключение

Предложенный и экспериментально исследованный в работе алгоритм автоматизированного построения сбалансированной декомпозиции, как видно из расчетов, позволяет достаточно эффективно реализовывать параллельное решение СЛАУ на системах с распределенной памятью. Сравнение прямых и итерационных методов решения вспомогательных СЛАУ в подобластях свидетельствует о преимуществе итерационных процессов в подпространствах Крылова, особенно при решении сверхбольших алгебраических систем. При этом решение подсистем на каждой внешней итерации достаточно проводить с невысокой точностью.

Работа поддержана грантом РФФИ № 14-07-00128, а также грантами Президиума РАН № 15.9-4 и ОМН РАН № 1.3.3-4.

Литература

1. Domain Decomposition Methods. URL: `http://ddm.org` (дата обращения: 14.03.2012)
2. 22nd International Conference on Domain Decomposition Methods (DD22). URL: `http://dd22.ics.usi.ch/` (дата обращения: 31.11.2013)
3. Intel (R) Math Kernel Library from Intel. URL: `http://software.intel.com/en-us/articles/intel-mkl/` (дата обращения : 08.04.2014)
4. Ильин, В.П. Методы и технологии конечных элементов / В.П. Ильин — Новосибирск: Изд-во ИВМиМГ СО РАН, 2007. — 371 с.
5. Писсанецки, С. Технология разреженных матриц / С. Писсанецки — М.: Мир, 1988. — 305 с.
6. Bramble, J.H. Convergence estimates for product iterative methods with applications to domain decomposition / J. Bramble, J. Pasciak, J. Wang, J. Xu // Mathematics of

Computation. — 1991. — V. 57, № 195. — P. 1–21.

7. Берж К. Теория графов и ее применения / К. Берж — М.: Изд-во иностр. лит., 1962. — 250 с.
8. Saad, Y. Iterative Methods for Sparse Linear Systems, Second Edition / Y. Saad — SIAM, 2003. — 528 p.
9. Кластер НКС-30Т. URL: `http://www2.sscc.ru/HKC-30T/HKC-30T.htm` (дата обращения: 12.02.2013).
10. Andreeva, M.Yu. Two solvers for nonsymmetric SLAE / M.Yu. Andreeva, V.P. Il'in, E.A. Itskovich // Bulletin NCC, «Numerical Analysis» series. — 2003. — Iss. 12. — P. 1–16.
11. Бутюгин, Д.С. Методы параллельного решения СЛАУ на системах с распределенной памятью в библиотеке Krylov / Д.С. Бутюгин, В.П. Ильин, Д.В. Перевозкин // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». — 2012. — Т. 47, № 306. — С. 5–19.

Валерий Павлович Ильин, д.ф.-м.н., профессор, главный научный сотрудник, Институт вычислительной математики и математической геофизики СО РАН (Новосибирск, Российская Федерация), ilin@sscc.ru.

Данил Валерьевич Перевозкин, младший научный сотрудник, Институт вычислительной математики и математической геофизики СО РАН (Новосибирск, Российская Федерация), foxillys@gmail.com.

Поступила в редакцию 10 апреля 2014 г.

Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 2, pp. 5–19

ON SOME VARIANTS OF DOMAIN DECOMPOSITION METHODS

V.P. Il'in, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation)**,**
D.V. Perevozkin, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation)

The paper considers the algorithms for solving large sparse SLAEs arising from grid approximations of boundary value problems. The SLAEs and algorithms are not limited in a sense of number of unknowns, computational nodes, processors and/or cores. This problem is reduced to a distributed variant of algebraic 3D-domain decomposition, in which no excessive load of the root process is present, i.e. all MPI-processes, each of which corresponds to its own subdomain, are almost equal. The computational process consists of two main stages. The first stage is the automatic decomposition, based on the analysis of the matrix portrait and the formation of large-block representation of the original SLAE. The second stage implements a Krylov subspace iterative process with FGMRes (flexible generalized minimal residual method) using either exact or approximate inverse of diagonal blocks as a preconditioner. The methods described are implemented as a part of Krylov, a library of algebraic solvers. The paper presents some features of current parallel implementation and estimates of resource usage. Efficiency of the developed algorithms is illustrated by solving several typical model problems with different parameters and in different configurations of multiprocessor computer systems.

Keywords: domain decomposition methods, matrix graphs, parallel algorithms, grid equations, sparse linear algebraic systems

References

1. Domain Decomposition Methods. URL: `http://ddm.org` (accessed: 14.03.2012)
2. 22nd International Conference on Domain Decomposition Methods (DD22). URL: `http://dd22.ics.usi.ch/` (accessed: 31.11.2013)
3. Intel (R) Math Kernel Library from Intel. URL: `http://software.intel.com/en-us/articles/intel-mkl/` (accessed: 08.04.2014)
4. Il'in V.P. Metody i tekhnologii konechnykh elementov [Methods and technologies of finite elements]. Novosibirsk, ICM&MG SBRAS Publishing, 2007.
5. Pissanetski S. Tekhnologiya razrezhennykh matrits [Sparse matrix technology]. Moscow, Mir Publishing, 1988.
6. Bramble, J.H. Convergence estimates for product iterative methods with applications to domain decomposition / J. Bramble, J. Pasciak, J. Wang, J. Xu // Mathematics of Computation. — 1991. — V. 57, № 195. — P. 1–21.
7. Berzh K. Teoriya grafov i eye primeneniya [Graph theory and its applications]. Moscow: Foreign Literature Publishing, 1962.

8. Saad, Y. Iterative Methods for Sparse Linear Systems, Second Edition / Y. Saad — SIAM, 2003. — 528 p.

9. NKS-30T cluster. URL: `http://www2.sscc.ru/HKC-30T/HKC-30T.htm` (accessed: 12.02.2013).

10. Andreeva, M.Yu. Two solvers for nonsymmetric SLAE / M.Yu. Andreeva, V.P. Il'in, E.A. Itskovich // Bulletin NCC, «Numerical Analysis» series. — 2003. — Iss. 12. — P. 1–16.

11. Butyugin D.S., Ilin V.P., Perevozkin D.V. Metody parallelnogo resheniya SLAU na sistemakh s raspredelennoy pamyatju v biblioteke Krylov [Parallel SLAE solution methods for distributed memory systems in Krylov library] Vestnik Yuzho-Uralskogo gosudarstvennogo universiteta. Seriya "Vychislitelnaya matematika i informatika"[Bulletin of South Ural State University. Series: Computational Mathematics and Informatics]. 2012. No. 47(306). P. 5–19.

Received 10 April 2014.

УДК 004.272, 532.517.4

ПРИМЕНЕНИЕ ГИБРИДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ К РЕШЕНИЮ ПРИКЛАДНЫХ ЗАДАЧ АЭРОГАЗОДИНАМИКИ[1]

Е.Ю. Карцева, А.В. Кашковский, А.О. Лацис, А.В. Панасенко, Ю.П. Смольянов, В.В. Чернов, А.А. Шершнёв

Использованию графических ускорителей при решении прикладных задач уделяется большое внимание. В работе рассмотрены возможности использования гибридной вычислительной системы, основу которой составляют универсальные и графические процессоры.

Ключевые слова: гибридная вычислительная система, GPGPU, прямое численное моделирование, сверхзвуковой свободный сдвиговый слой.

Введение

Создание современных изделий ракетно-космической техники требует проведения имитационного моделирования их поведения в реальных условиях. Такие вычисления требуют использования высокопроизводительных вычислительных систем значительно более мощных, чем, например, рабочие станции. Поэтому приоритетным направлением является создание кластерных вычислительных систем с гибридной архитектурой на основе использования графических ускорителей, что является одним из самых недорогих и эффективных путей решения проблемы. Наиболее распространены в настоящее время графические ускорители компании NVIDIA [1].

1. Гибридная кластерная вычислительная система

В 2013 г. в ФГУП ЦНИИмаш была установлена гибридная кластерная вычислительная система (ГКВС), разработанная в ИПМ им. М.В. Келдыша РАН при участии ФГУП ЦНИИмаш. Кластер состоит из управляющего и 6 вычислительных узлов, соединенных с помощью коммутационной среды «МВС-Экспресс». Каждый вычислительный узел содержит по два 8-ядерных процессора Intel Xeon E7-8837 8-Cores и 3 графических ускорителя NVIDIA Tesla K20. Для ГКВС были адаптированы комплексы программ моделирования аэрогазодинамики.

Коммутационная среда «МВС-Экспресс» основана на стандарте компьютерной шины PCI Express (PCIe) и имеет древовидную структуру. Корнем дерева является ведущий узел, или Root Complex (в терминах шины PCIe). Листьями дерева являются устройства, подключенные к шине, таковыми здесь являются интерфейсные платы других вычислительных модулей, подключенных через коммутатор. Вершины дерева, не являющиеся листьями, представляют собой коммутаторы, т.е. устройства, способные принять пакет по некоторому входу, определить в пакете адрес назначения и в соответствии с ним выдать пакет на указанный выход.

Сеть «МВС-Экспресс», состоящая из коммутатора и интерфейсных плат, является совместной разработкой ИПМ им. М.В. Келдыша РАН и ФГУП «НИИ «Квант».

[1] Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии – 2014».

Кластер, построенный по технологии прямой коммутации каналов PCI-Express, реализует на аппаратном уровне общее поле памяти большого объема и внешних устройств для всех входящих в кластер узлов. При этом уровни скоростей и задержек при передаче данных в такой сети сравнимы с аналогичными показателями для передачи данных внутри компьютера. Так, время задержки при единичной записи данных в «чужую» память сокращается с 3-6 мкс до 200-400 нс.

2. Результаты оценки производительности кластера

В целях минимизации обмена данными между CPU и GPU расчет полностью проводится на GPU. При использовании GPU каждое ядро обычного процессора передает свою подобласть на отдельный графический ускоритель. В ходе расчета происходит лишь обмен значениями параметров граничных ячеек, как показано на рис. 1.

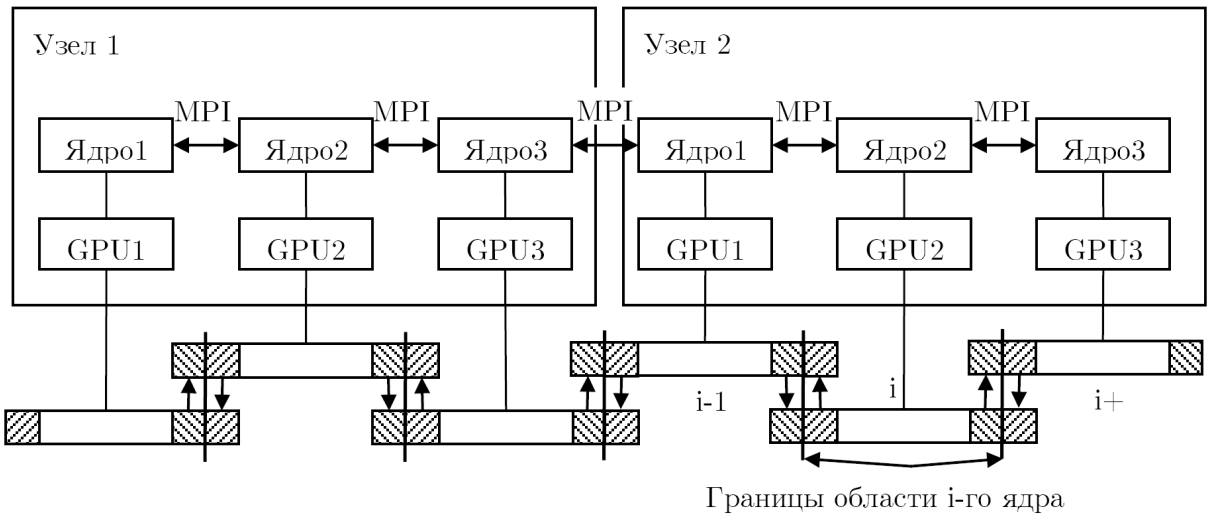


Рис. 1. Структура вычислений на CPU – GPU

Результаты оценки производительности гибридного кластера приведены в таблицах 1, 2 на примерах решения типовой задачи, возникающей при разработке ракетно-космической техники, – расчета аэродинамики спуска космического корабля типа «Союз» и результатах прямого расчета неустойчивости в сдвиговом слое программой ns_cu, рассмотренной ниже.

Таблица 1

Время расчета и ускорение

N	Время расчета, с		Ускорение относительно N ядер CPU	Ускорение относительно 1 ядра CPU
	N ядер CPU	N GPU		
1	769,144	16,008	48,05	48,05
2	396,388	8,277	47,89	92,92
3	266,207	5,954	44,71	129,17
6	135,277	3,716	36,41	206,99
12	64,590	2,730	23,66	281,77

В таблице 1 приведено время расчета и ускорение при расчете обтекания спускаемого аппарата космического корабля типа «Союз» в рамках модели уравнений Эйлера.

Использовалась расчетная сетка размером 768×128×16 ячеек. Расчеты проводились на различном количестве GPU и для сравнения на том же количестве ядер CPU.

В таблице 2 представлены времена расчетов неустойчивости в сдвиговом слое с помощью программы ns_cu при разных количествах и конфигурациях используемых GPU, а также ускорения и эффективности относительно расчета на одном GPU. На рис. 2 приведены графики ускорения и эффективности.

Таблица 2

Времена выполнения одинакового количества шагов программы ns_cu при разных количествах GPU

Количество GPU	**Конфигурация**	**Время расчета, сек**	**Ускорение по сравнению с 1 GPU**	**Эффективность**
1	1 узел с 1 GPU	2394,00	1,00	1,00
2	1 узел с 2 GPU	1200,00	1,99	0,997
3	1 узел с 3 GPU	843,00	2,84	0,95
3	3 узла по 1 GPU	822,00	2,91	0,97
6	2 узла по 3 GPU	407,00	5,88	0,98
6	3 узла по 2 GPU	413,00	5,80	0,97
6	6 узлов по 1 GPU	414,00	5,78	0,96
9	3 узла по 3 GPU	286,00	8,37	0,93
12	4 узла по 3 GPU	222,00	10,78	0,90
15	5 узлов по 3 GPU	181,00	13,23	0,88
18	6 узлов по 3 GPU	155,00	15,45	0,86

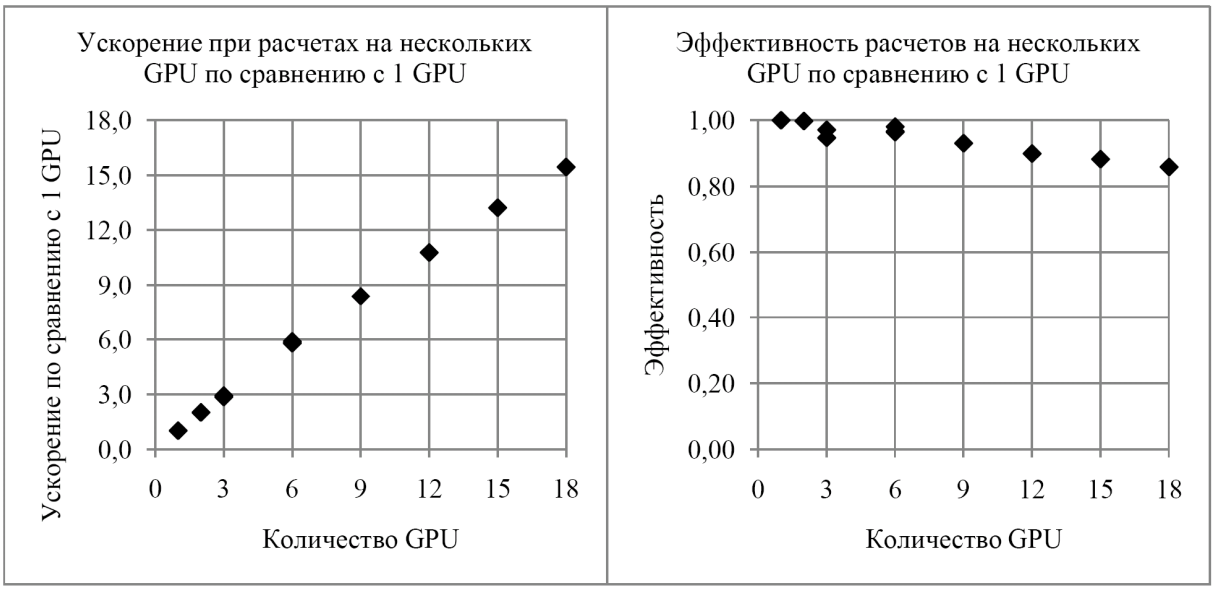


Рис. 2. Ускорение и эффективность программы ns_cu при различном количестве GPU

Можно отметить высокую эффективность распараллеливания. Расчет на одном узле с тремя GPU идет медленнее расчета на трех узлах с одним GPU на каждом узле.

3. Пример расчета

Моделирование развития неустойчивости в свободном сдвиговом слое (задача о слое смешения) в двумерной постановке является одним из простых случаев моделирования турбулентности. Моделирование данной задачи проводилось на основе прямого численного решения полных нестационарных уравнений Навье-Стокса с использованием программы ns_cu, разработанной в ИТПМ СО РАН. Схема течения показана на рис. 3. Рассматривалось течение, образующееся при смешении двух параллельных сверхзвуковых потоков газа, направленных вдоль оси Z и имеющих в начальный момент времени плоскую границу раздела, перпендикулярную оси X. В таком течении на границе потоков возникает неустойчивость Кельвина-Гельмгольца, приводящая к переходу ламинарного слоя смешения в турбулентный.

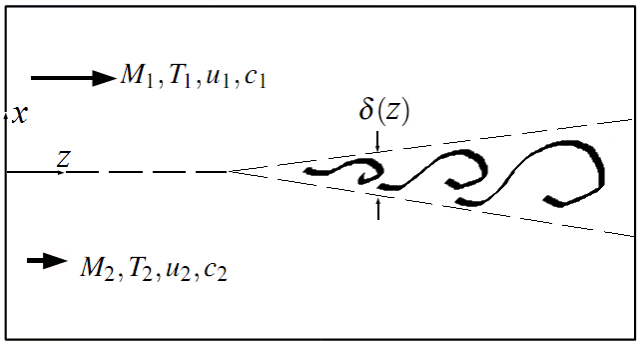


Рис. 3. Схема течения

На входной границе расчетной области задавался «ступенчатый» профиль основного течения (таким образом, начальная толщина слоя смешения равняется нулю) с наложенными на него возмущениями поперечной скорости U_x, зависящими от времени. В безразмерном виде, используемом в расчете (линейный размер в задаче отсутствует), параметры отнесены к их значениям в более быстром потоке: скорости обезразмерены на скорость звука, плотности и температуры – на плотность и температуру. Возмущения поперечной скорости имели вид $A\cos(\omega t - \alpha x)e^{-x^2/\sigma^2}$. На боковых гранях величины экстраполировались с нулевым порядком (снос потока). Число Рейнольдса, определяемое по параметрам более быстрого потока и единице длины, равнялось 1000; число Рейнольдса, определенное по параметрам более быстрого потока и длине расчетной области, равнялось $2{,}7\cdot 10^6$.

Для вычислений на нескольких GPU вся расчетная область разделялась на число подобластей, равное числу GPU, и каждый GPU производил вычисления в своей подобласти и периодически обменивается данными с другими GPU. Для управления несколькими GPU, установленными на одном узле, использовалась технология OpenMP, а обмен данными на этом узле осуществлялся через память GPU. Для обмена данными между узлами использовался протокол MPI.

Ниже представлены результаты двух расчетов, параметры которых описаны в таблицах 3-5.

В таблицах 3 и 4 даны размерные и безразмерные начальные данные. Размерные величины обозначены индексом «*», обезразмеренные – тем же символом без индекса. Обозначения: C_1^*, C_2^* – скорости звука в потоках ($C^* = 20{,}1\sqrt{T^*}$);

U_1^*, U_2^* – скорости первого и второго потоков ($U_1^* > U_2^*$); $U_1 = U_1^* / C_1^*$; $U_2 = U_2^* / C_1^*$;

T_1^*, T_2^* – температуры первого и второго потоков; $T_1 = 1$; $T_2 = T_2^* / T_1^*$;

ρ_1^*, ρ_2^* – плотности потоков; $\rho_1 = 1$; $\rho_2 = \rho_2^* / \rho_1^*$;

$r = \dfrac{U_2^*}{U_1^*} = \dfrac{U_2}{U_1}$; $s = \dfrac{\rho_2^*}{\rho_1^*} = \dfrac{\rho_2}{\rho_1}$; $\lambda_s = \dfrac{(1-r)(1+\sqrt{s})}{2(1+r\sqrt{s})}$;

P^* – давление (одинаковое для обоих потоков);

M_1, M_2 – числа Маха первого и второго потоков;

$M_c = (U_1^* - U_2^*)/(C_1^* + C_2^*)$ – конвективное число Маха.

Таблица 3

Начальные данные

№ расчета	U_1^*, м/с	U_2^*, м/с	r	λ_s	P^*, Па	T_1^*, К	T_2^*, К	M_1	M_2	M_c
1	1000	400	0,4	0,43	10^5	300	300	2,872	1,1488	0,86
2	1000	600	0,6	0,25	10^5	300	300	2,872	1,7232	0,57

Таблица 4

Начальные данные (продолжение)

№ расчета	U_1	U_2	T_1	T_2	ρ_1	ρ_2	Возмущения на входной границе			
							A	a	ω	σ
1	2,872	1,1488	1	1	1	1	0,05	1,987	3,995	1
2	2,872	1,7232	1	1	1	1	0,05	0,971	0,777	1

В таблице 5 перечислены такие параметры расчета, как число ячеек расчетной сетки, размер расчетной области, число GPU, безразмерное время окончания расчета, время выполнения расчета. В расчетах использовалась равномерная расчетная сетка.

Таблица 5

Параметры расчета

№ расчета	Число ячеек расчетной сетки, $N_x \times N_y \times N_z$	Размер расчетной области, $L_x \times L_y \times L_z$	Число GPU	Безразмерное время окончания расчета	Время выполнения расчета, часов
1	600×3×2700	600,0×3,0×2700,0	12	1600	8
2	500×3×2700	500,0×3,0×2700,0	9	1700	9,3

На рис. 4 показаны мгновенные картины распределений продольной скорости, энтропии и усредненная по времени картина распределения продольной скорости для расчета № 1. На рис. 5 показаны эти же картины для расчета № 2. Все данные приведены в безразмерном виде, используемом в расчете. На рисунках показаны границы блоков, считающихся на разных GPU.

На рис. 6 показаны нормализованные профили усредненных продольных скоростей в сравнении с функцией вида гиперболического тангенса в координатах (x^*, U^*), где $x^* = (x - x_{0,5})/\delta_\omega$; $U^* = (U - U_2)/(U_1 - U_2)$; $x_{0,5}$ – значение x, в котором $U^* = 0{,}5$; δ_ω – толщина завихренности, определяемая по максимальному наклону профиля средней скорости: $\delta_\omega = \dfrac{U_1 - U_2}{\left[\partial U / \partial x\right]_{\max}}$. Как и описано, например, в [2,3], наблюдается хорошее соответствие кривых.

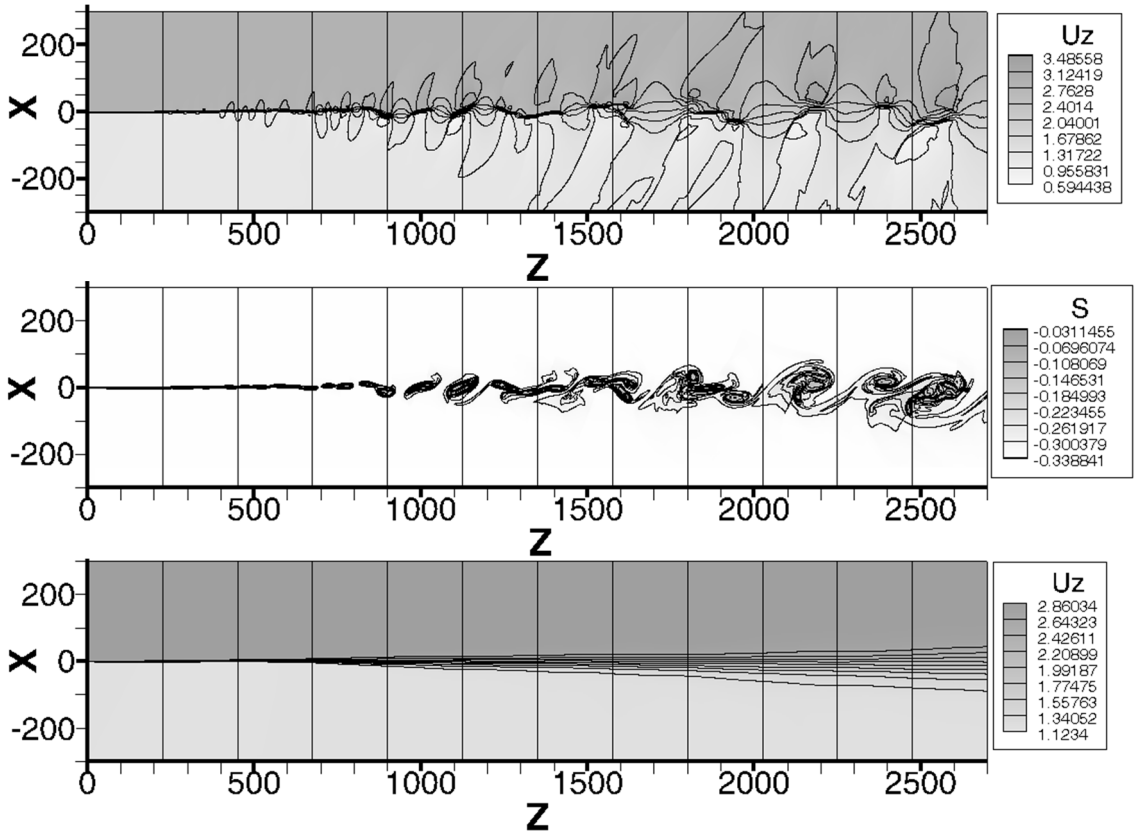


Рис. 4. Мгновенные картины распределений продольной скорости и энтропии и усредненная по времени картина распределения продольной скорости для расчета № 1 ($M_c = 0{,}86$)

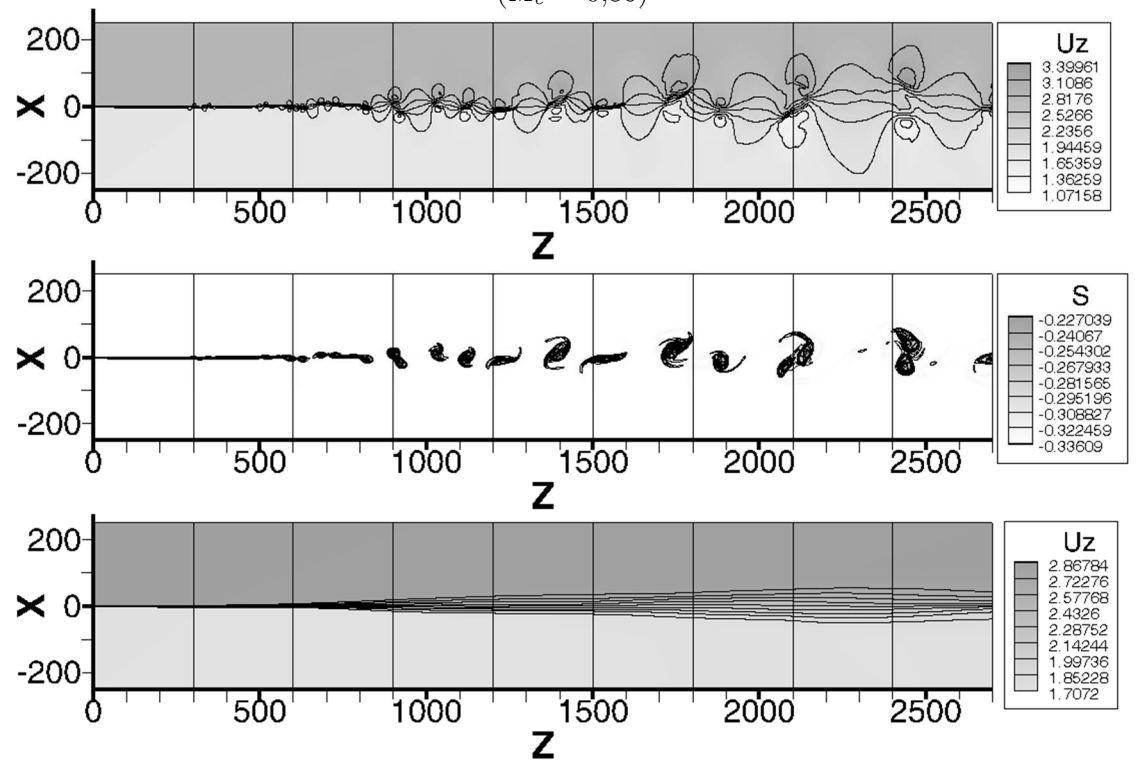


Рис. 5. Мгновенные картины распределений продольной скорости и энтропии и усредненная по времени картина распределения продольной скорости для расчета № 2 ($M_c = 0{,}57$)

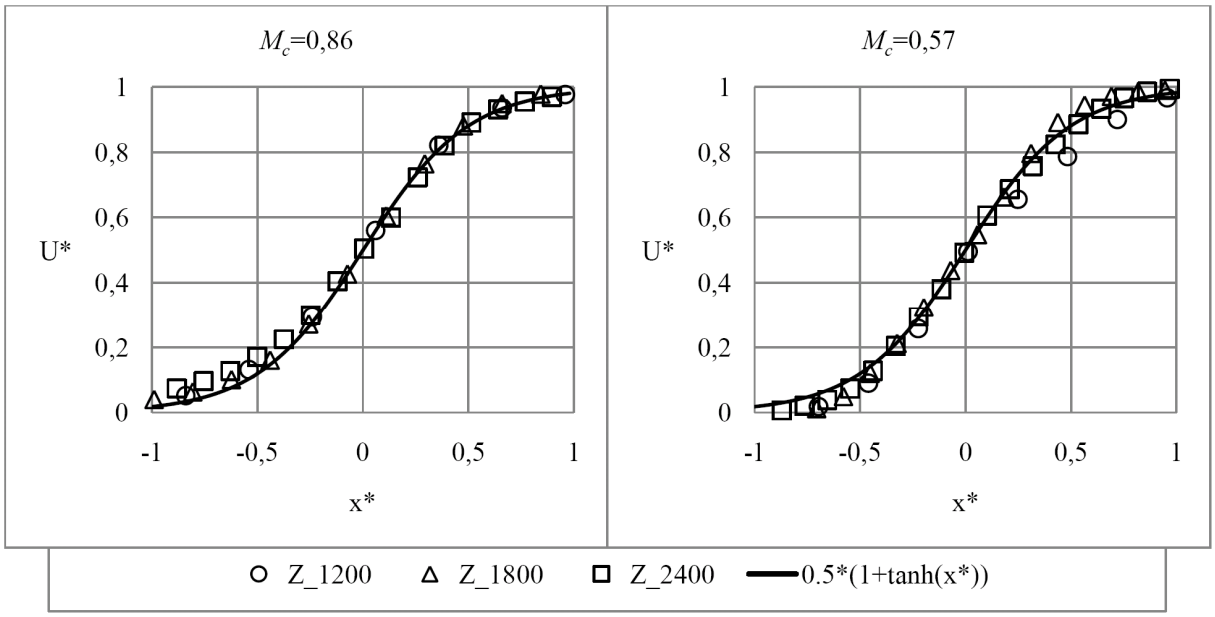


Рис. 6. Профили продольной скорости в различных сечениях:
слева – расчет № 1 ($M_c = 0{,}86$), справа – расчет № 2 ($M_c = 0{,}57$)

На рис. 7 представлена толщина слоя смешения в зависимости от расстояния от входной границы для расчетов № 1 и № 2. Показаны две различные толщины слоя смешения, определяемые из усредненных значений поля скоростей: определенная выше толщина завихренности δ_ω, а также δ_b – так называемая 10 % толщина по скорости, определяемая как толщина слоя, внутри которого скорость отличается от скоростей начальных потоков более чем на 10 % от разницы этих скоростей. Как известно [4], толщина турбулентного слоя смешения в некотором сечении пропорциональна расстоянию от точки начала турбулентности до этого сечения. Таким образом, вводится такой важный параметр, как скорость расширения слоя смешения. По линейной части графиков рис. 7 были определены значения $d\delta_b / dz$ скоростей расширения слоя смешения для 10 % толщины по скорости: 0,0485 для $M_c = 0{,}86$ и 0,0467 для $M_c = 0{,}57$.

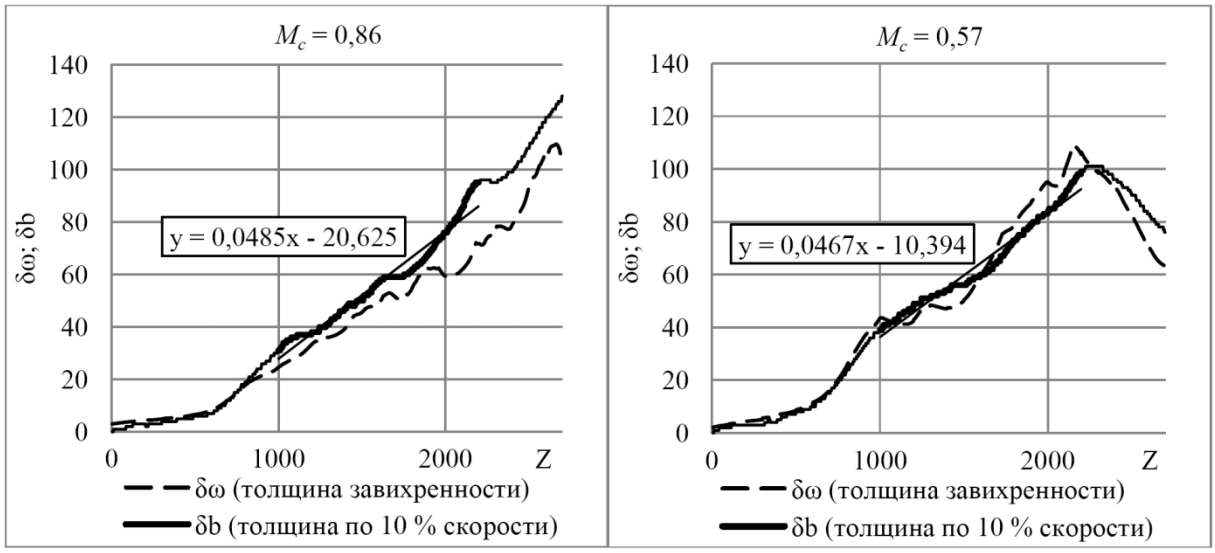


Рис. 7. Толщины слоя смешения в зависимости от расстояния от входной границы:
слева – расчет № 1 ($M_c = 0{,}86$), справа – расчет № 2 ($M_c = 0{,}57$)

Известно [4], что скорость расширения сверхзвукового слоя смешения меньше, чем при несжимаемом течении. По методике, подробно описанной в [2], на основе 10 % толщины по скорости для каждого расчета был вычислен коэффициент Ф, определенный как отношение скорости расширения сжимаемого слоя смешения к скорости роста несжимаемого слоя смешения при тех же отношениях скоростей и температур:

$\Phi = \dfrac{d\delta_b / dz}{(d\delta_b / dz)_i}$, где $(d\delta_b / dz)_i$ – скорость роста несжимаемого слоя смешения.

Согласно [2], скорость расширения несжимаемого слоя смешения, определенного по 10% толщине по скорости, вычисляется как $(d\delta_b / dz)_i = E_b \lambda_s$, где E_b=0,200.

Таким образом, имеем для расчета № 1 ($M_c = 0{,}86$) $\Phi = 0{,}56$, для расчета № 2 ($M_c = 0{,}57$) $\Phi = 0{,}93$.

На рис. 8 показано сравнение вычисленных значений коэффициента Ф с экспериментальными данными, собранными в [2], и расчетными данными из [5]. Видно хорошее совпадение расчетных данных с ранее известными.

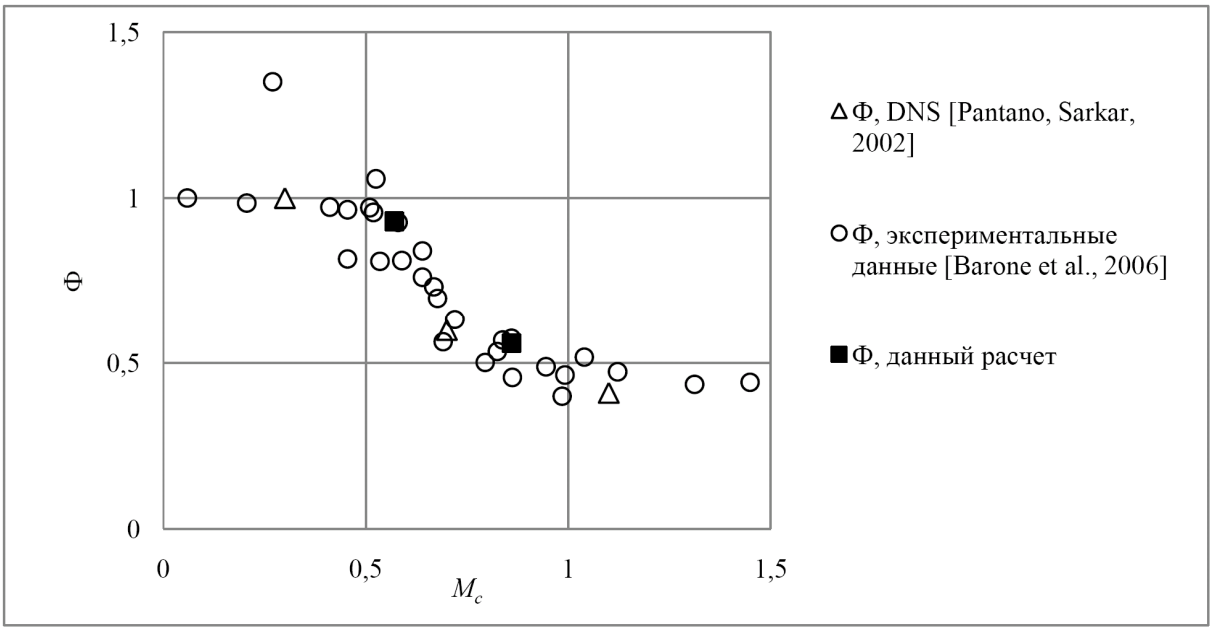


Рис. 8. Отношение скорости расширения сжимаемого слоя смешения к скорости роста несжимаемого слоя смешения в зависимости от конвективного числа Маха

Заключение

Проведена проверка производительности гибридного кластера, в состав которого входят классические процессоры и графические ускорители NVIDIA Tesla K20. В расчетах достигнуто ускорение на графических ускорителях до 23 раз при использовании 12 графических ускорителей по отношению к 12 ядрам обычного процессора.

Прямое моделирование турбулентного смешения двух газодинамических потоков показало возможность расчета при использовании гибридного кластера с соответствием имеющимся результатам других авторов.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №13-08-12033.

Литература

1. Сайт компании NVIDIA. URL: http://www.nvidia.com (дата обращения: 05.02.2014).
2. Barone, M.F. Validation case study: Prediction of compressible turbulent mixing layer growth rate. / M.F. Barone, W.L. Oberkampf, F.G. Blottner // AIAA Journal. — 2006. — Vol. 44, No. 7. — P. 1488–1497.
3. Samimy, M. Compressible mixing layer with and without particles. — Technical Report MEMS-92-101. — Ohio State University, 1992. / M. Samimy, G.S. Elliott, D.D. Glawe, M.F. Reeder, S.A. Arnette.
4. Brown, G.L. On density effects and large structure in turbulent mixing layers / G.L. Brown, A. Roshko // Journal of Fluid Mechanics. — 1974. — Vol. 64, No. 4. — P. 775–816.
5. Pantano, C. Study of Compressibility Effects in the High-Speed Turbulent Shear Layer Using Direct Simulation / C. Pantano, S.A Sarkar // Journal of Fluid Mechanics. — 2002. — Vol. 451. — P. 329–371.

Карцева Елена Юрьевна, инженер отдела, Федеральное государственное унитарное предприятие «Центральный научно-исследовательский институт машиностроения» (Королёв, Российская Федерация), kartsevaelena@mail.ru.

Кашковский Александр Владимирович, к.т.н., научный сотрудник, Институт теоретической и прикладной механики им. С.А. Христиановича Сибирского отделения Российской академии наук (Новосибирск, Российская Федерация), sasa@itam.nsc.ru

Лацис Алексей Оттович, д.ф.-м.н, заведующий сектором нетрадиционных суперкомпьютерных архитектур, Институт прикладной математики им. М.В. Келдыша Российской академии наук (Москва, Российская Федерация)

Панасенко Александр Викторович, к.т.н., с.н.с., начальник отдела, Федеральное государственное унитарное предприятие «Центральный научно-исследовательский институт машиностроения» (Королёв, Российская Федерация), akpanas@mail.ru

Смольянов Юрий Павлович, к.ф.-м.н., начальник отдела вычислительных систем и локальных сетей, Институт прикладной математики им. М.В. Келдыша Российской академии наук (Москва, Российская Федерация)

Чернов Владимир Владимирович, ведущий инженер отдела, Федеральное государственное унитарное предприятие «Центральный научно-исследовательский институт машиностроения» (Королёв, Российская Федерация), chernov-vv@mail.ru

Шершнёв Антон Алексеевич, к.ф.-м.н., младший научный сотрудник, Институт теоретической и прикладной механики им. С.А. Христиановича Сибирского отделения Российской академии наук (Новосибирск, Российская Федерация), antony@itam.nsc.ru

Поступила в редакцию 6 марта 2014 г.

Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 2, pp. 20–29

APPLYING HETEROGENEOUS COMPUTING SYSTEM FOR SOLVING APPLIED PROBLEMS OF AEROGASDYNAMICS

E.Yu. Kartseva, Central Research Institute of Machine Building (Korolyov, Russian Federation),
A.V. Kashkovsky, Khristianovich Institute of Theoretical and Applied Mechanics (Novosibirsk, Russian Federation),
A.O. Latsis, Keldysh Institute of Applied Mathematics (Moscow, Russian Federation),
A.V. Panasenko, Central Research Institute of Machine Building (Korolyov, Russian Federation),
Yu.P. Smolyanov, Keldysh Institute of Applied Mathematics (Moscow, Russian Federation),
V.V. Chernov, Central Research Institute of Machine Building (Korolyov, Russian Federation),
A.A. Shershnyov, Khristianovich Institute of Theoretical and Applied Mechanics (Novosibirsk, Russian Federation)

Using graphics accelerators for solving applied problems is emphasized. This paper shows usability of heterogeneous computing system consisting of central processing units and graphics processing units.

Keywords: heterogeneous computing system, GPGPU, direct numerical simulation, supersonic free shear layer.

References

1. NVIDIA Corporation website. URL: http://www.nvidia.com (accessed: 05.02.2014).
2. Barone M.F., Oberkampf W.L., Blottner F. G. Validation case study: Prediction of compressible turbulent mixing layer growth rate // AIAA Journal. 2006. Vol. 44, No. 7. P. 1488–1497.
3. Samimy M., Elliott G.S., Glawe D.D., Reeder M.F., Arnette S.A. Compressible mixing layer with and without particles. Technical Report MEMS-92-101. Ohio State University, 1992.
4. Brown G.L., Roshko A. On density effects and large structure in turbulent mixing layers // Journal of Fluid Mechanics. 1974. Vol. 64, No. 4. P. 775–816.
5. Pantano C., Sarkar S.A. Study of Compressibility Effects in the High-Speed Turbulent Shear Layer Using Direct Simulation // Journal of Fluid Mechanics. 2002. Vol. 451. P. 329–371.

Received 6 March 2014.

УДК 004.457

МЕТОДЫ И СРЕДСТВА МЕТАМОНИТОРИНГА РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕД[1]

И.А. Сидоров, А.П. Новопашин, Г.А. Опарин, В.В. Скоров

Работа посвящена решению проблемы мониторинга распределенной вычислительной среды, состоящей из множества слабосвязанных разнородных вычислительных ресурсов. Главной отличительной особенностью и аспектом новизны разработанных методов и средств является использование универсальных программных агентов, способных осуществлять сбор данных от локальных систем мониторинга различных компонентов среды, унифицировать и анализировать эти данные, а также при необходимости вырабатывать и исполнять в автоматическом режиме управляющие воздействия, направленные на изменение режимов работы оборудования, в том числе его отключение при возникновении аварийных ситуаций. В составе каждого агента системы метамониторинга реализованы: подсистема сбора данных, подсистема взаимодействия с СУБД, экспертная подсистема и управляющая подсистема. Подход основан на применении веб-технологий, мультиагентных технологий, методов создания экспертных систем, методов децентрализованной обработки и распределенного хранения данных.

Ключевые слова: распределенные вычислительные среды, мониторинг, мультиагентные технологии, экспертные системы

Введение

Рассматриваемые в статье распределенные вычислительные среды (РВС) предназначены для проведения ресурсоемких вычислительных экспериментов и включают высокопроизводительные системы различной архитектуры и конфигурации: выделенные серверы, гибридные вычислительные системы с модулями ускорения вычислений на базе графических ускорителей, сопроцессоров и ПЛИС, вычислительные кластеры (суперкомпьютеры), функционирующие под управлением различных систем управления прохождением заданиями (PBS/Torque, Cleo, СУППЗ.

При организации таких РВС особое внимание уделяется средствам управления и мониторинга. Основными функциями средств управления РВС является распределение нагрузки по вычислительным узлам, реализация непараллельных и параллельных команд, запуск и остановка вычислительных узлов и ряд других. Назначение средств мониторинга состоит в обеспечении оператора РВС структурированной и унифицированной информацией о загрузке ресурсов вычислительных узлов (процессоров, оперативной памяти, системы ввода-вывода, коммуникационных интерфейсов), физическом состоянии вычислительного оборудования (температуре процессоров и материнских плат), работе устройств инженерной инфраструктуры (кондиционеров, чиллеров, систем бесперебойного питания) и текущем состоянии окружающей среды.

С ростом числа вычислительных узлов в составе РВС увеличивается объем данных, которые приходится анализировать оператору. Как следствие, возрастает вероятность возникновения ошибок, обусловленных человеческим фактором. В связи с этим, в больших РВС огромное значение имеет форма представлениях данных мониторинга, их

[1] Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии – 2014»

наглядность, информативность и актуальность – свойства, позволяющие обслуживающему персоналу РВС оперативно выявлять причины неисправностей и сбоев, своевременно реагировать на изменения в работе системы.

Известные на сегодняшний день средства мониторинга имеют ряд недостатков и ограничений, не позволяющих эффективно применять их в сложных вычислительных системах. В частности, в популярной системе Ganglia отсутствует возможность реализации механизмов обработки текущих значений и уведомлений о событиях. Две другие часто используемые системы – Nagios и Zabbix лишены этого недостатка, однако их первоначальная ориентация на мониторинг сетевого оборудования делает их малоприспособленными для РВС.

Следует выделить отдельно систему мониторинга LAPTA [1], предназначенную для многоаспектного анализа динамических характеристик параллельных программ, выполняемых на суперкомпьютерах. Система предполагает тесную интеграцию с известными средствами управления заданиями. Сбор информации о ресурсах осуществляется с привязкой к конкретной задаче пользователя, что представляется, с нашей точки зрения, наиболее верным направлением развития систем мониторинга вычислительных кластеров и РВС. В то же время, зависимость компонентов системы мониторинга LAPTA от специализированных интерпретируемых языков программирования и систем хранения данных значительно усложняет ее установку и настройку. Кроме того, реализация агента системы мониторинга на основе интерпретируемого языка программирования требует значительных системных ресурсов, что может приводить к снижению эффективности работы пользовательских приложений на узлах РВС.

В данной работе предлагается подход к созданию инструментального комплекса метамониторинга РВС, отличающейся от известных уникальным сочетанием свойств, к числу которых относится: автоматический контроль программно-аппаратных ресурсов с использованием мультиагентных технологий, децентрализованная схема хранения данных метамониторинга, применение экспертных систем для принятия решений.

Исходные требования к разрабатываемому инструментальному комплексу метамониторинга РВС [2] предполагают, что он должен:

- базироваться на принципах организации мультиагентных систем;
- обеспечивать высокую степень автономности компонентов;
- иметь многоуровневую иерархическую структуру и средства, гарантирующие быстрое и надежное взаимодействие различных уровней, прямую и обратную связь между ними;
- обеспечивать возможность интеграции с популярными системами мониторинга высокопроизводительных ресурсов (Ganglia, Nagios и др.);
- предоставлять широкий набор функций сбора данных о состоянии программно-аппаратных компонентов вычислительных узлов, предоставлять средства разработки модулей сбора данных на различных языках системного программирования (C, Perl, Bash и др.);
- предоставлять средства сбора и анализа данных о состоянии оборудования вспомогательной (инженерной) инфраструктуры вычислительных кластеров;
- включать средства унификации данных, получаемых из различных источников;
- использовать технологию децентрализованного хранения данных, обеспечивающую снижение нагрузки на центральный узел системы метамониторинга;

– предоставлять средства автоматизированного экспертного анализа данных и генерации управляющий воздействий;
– включать средства визуализации, наглядно отображающие состояние ресурсов, задействованных в ходе решения задач;
– предоставлять прикладные программные интерфейсы на основе открытых стандартов для интеграции с другими программными комплексами.

1. Модель мультиагентной системы метамониторинга

Модель мультиагентной системы метамониторинга программно-аппаратных ресурсов (узлов) РВС можно представить в виде следующей структуры:

$$D =< N, M, T, A, R, S >,$$

где:
– N – множество узлов РВС;
– M – множество измеряемых метрик (характеристик) узлов РВС;
– T – множество технических средств коммуникации узлов РВС;
– A – множество агентов системы метамониторинга;
– R – множество правил вывода экспертной подсистемы;
– S – множество управляющих воздействий.

Связи между элементами множеств заданы отношениями:
– $NM \subseteq N \times M$ – отношение, определяющее множество метрик, измеряемых на узлах РВС;
– $NT \subseteq N \times T$ – отношение, определяющее топологию сети узлов РВС;
– $NA \subseteq N \times A$ – отношение, определяющее иерархическую структуру мультиагентной среды;
– $RM \subseteq R \times M$ – отношение, определяющее множество правил вывода, используемых экспертной подсистемой;
– $SM \subseteq S \times M$ – отношение, определяющее множество управляющих воздействий, генерируемых экспертной подсистемой по результатам анализа значений метрик.

Задача метамониторинга сформулирована следующим образом. Пусть $n_i \subset N$ – отдельный узел РВС. Для i-го узла измеряются m_i метрик ($m_i \subset M$). $B_i(t) = \{b_{ik}(t)\}$ – множество значений метрик для i-го узла в момент времени t, где $k = 1, 2, \ldots, m_i$.

Устойчивым состоянием метрик узла на отрезке времени $[t_0, t_1]$ считается такое, для которого выполняется неравенство:

$$| b_{ik}(t) - b_{ik}(t_0) | \le c_{ik}, \forall t \in [t_0, t_1], \forall k \in [1, m_i],$$

где c_{ik} – положительная константа, характеризующая допустимый разброс значений для каждой метрики.

Значимым событием является переход узла из устойчивого состояния $B_i(t_0)$ в состояние $B_i(t_1)$, при котором происходит существенное изменение значения какой-либо метрики из множества m_i.

При возникновении значимого события на отрезке времени $[t_0, t_1]$ формируется перечень фактов для экспертной подсистемы, и на основе правил вывода R_i, определенных для узла i, машиной вывода генерируются управляющие воздействия S_i.

Под управляющим воздействием S_i понимается инструкция, предназначенная для реагирования на значимое событие. Среди таких управляющих воздействий могут быть:
– уведомление оператора системы метамониторинга РВС;

– уведомление пользователя, задача которого выполняется на узле n_i;
– снятие задач пользователя, выполняющихся на узле n_i;
– выключение узла n_i (к примеру, при длительном простое или выявлении критических значений заданных метрик);
– включение узла n_i (к примеру, при появлении задач в очереди).

Сгенерированное экспертной подсистемой управляющее воздействие передается управляющей подсистеме, где оно транслируется в системные команды и затем исполняется.

Далее состояние узла (B_i) передается агентам верхних уровней для анализа и генерации управляющих воздействий в соответствии с правилами, определенными на соответствующих уровнях.

2. Архитектура мультиагентной системы метамониторинга

Архитектура мультиагентной системы метамониторинга РВС включает следующие основные компоненты (показано на рис. 1):

– средства доступа пользователей, позволяющие взаимодействовать с системой метамониторинга как в пакетном, так и в интерактивном режимах;
– подсистемы уровня доступа, осуществляющие контроль прав доступа к запрашиваемым данным и реализующие серверную часть графического интерфейса пользователя;
– агент верхнего уровня, функционирующий в центральном узле РВС и выполняющий основную задачу по управлению системой метамониторинга;
– агенты промежуточного уровня, функционирующие на промежуточных узлах и решающие задачу снижения нагрузки на агенты верхних уровней;
– агенты нижнего уровня, функционирующие на исполнительных узлах РВС и осуществляющие сбор и первичную обработку данных о состоянии узла;
– подсистема децентрализованного хранения данных, предоставляющая функции для работы с данными для агентов разных уровней.

Иерархическая структура представляемой системы метамониторинга достигается путем организации многоуровневой мультиагентной среды, возглавляемой агентом верхнего уровня, которому подчинены агенты промежуточного и нижнего уровня. Агенты промежуточного уровня могут осуществлять управление как группой агентов нижнего уровня, так и группой агентов дополнительных промежуточных уровней, которых в системе может быть несколько. Такая иерархия позволяет обеспечить высокую масштабируемость системы метамониторинга для РВС, состоящей из большого числа узлов.

Каждый агент мультиагентной системы метамониторинга реализован в виде программы, функционирующей в фоновом режиме, и обладает свойствами, характерными для автономных программных агентов [3], а именно, реактивностью, автономностью, целенаправленностью и коммуникативностью.

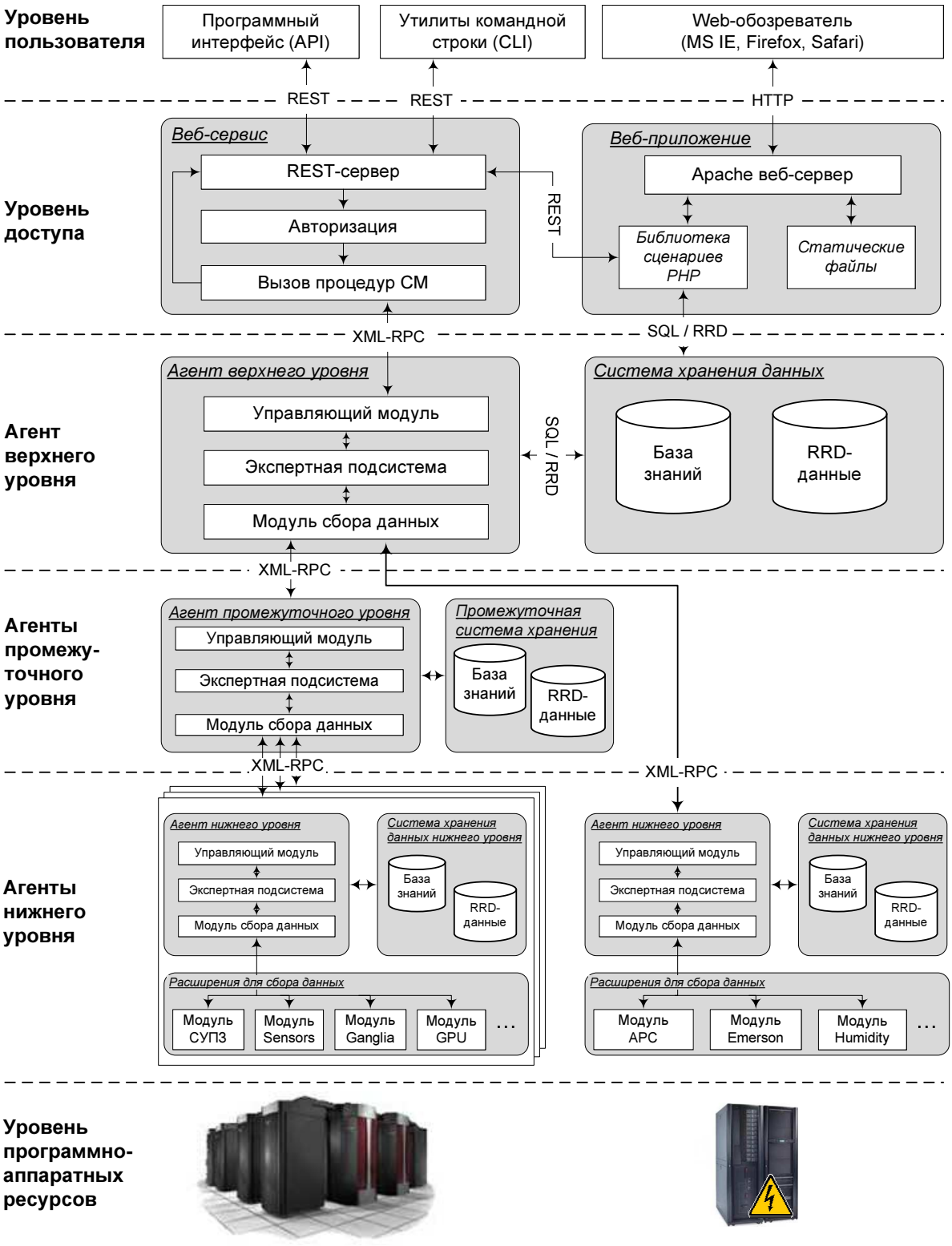


Рис. 1. Архитектура системы метамониторинга РВС

В составе агента системы метамониторинга реализованы следующие подсистемы:

– подсистема сбора данных, осуществляющая считывание показаний программных и аппаратных сенсоров различных устройств, прием данных от локальных систем мониторинга и агентов разных уровней, унификацию полученных данных и их трансляцию в промежуточный формат подсистемы взаимодействия с СУБД;

– подсистема взаимодействия с СУБД, выполняющая функции агрегации и контроля целостности данных;
– экспертная подсистема, выполняющая функции анализа данных, полученных за определенный интервал времени, и генерации управляющих воздействий на основе проведенного анализа;
– управляющая подсистема, реализующая функции исполнения управляющих воздействий и взаимодействия с агентами верхних уровней.

Агент нижнего уровня устанавливается непосредственно на узле РВС и выполняет функции сбора данных о состоянии узла, анализа этих данных, генерации и исполнения управляющий воздействий, а также функции взаимодействия с другими агентами.

Возможность анализа данных и принятия необходимых решений на стороне узла является ключевой особенностью реализуемого подхода. Как правило, клиенты систем мониторинга выполняют лишь функции сбора и периодической отправки данных в центральный узел, где производится обработка и анализ этих данных. При такой реализации существенно снижается количество операций по анализу данных на стороне узла, однако создается дополнительная нагрузка на стек сетевых протоколов, обработка которого (формирование, разбор и контроль сетевых пакетов) также требует процессорного времени. В нашем подходе анализ данных на стороне узла позволяет производить пересылку данных на центральный узел только в случае необходимости (либо по специальному запросу агентов верхних уровней). Установлено, что первичный анализ данных на стороне узла требует значительно меньше ресурсов, чем затраты на формирование сетевых пакетов и проверки их целостности. Сравнительный анализ работы агентов системы мониторинга Ganglia (gmond) и агентов разрабатываемой системы показал, что при одинаковой периодичности опроса датчиков (каждые 30 сек.), последний потребляет на 37% процессорного времени меньше. Кроме того, базовый набор сенсоров разрабатываемого агента значительно шире набора сенсоров агента системы мониторинга Ganglia. Таким образом, помимо снижения нагрузки на сеть и центральный узел системы метамониторинга удается снизить негативное влияние агента на производительность решения задач на узлах РВС.

Сбор информации о состоянии узла и загрузке отдельных его компонентов (процессора, памяти, сетевых интерфейсов и др.) реализован с использованием функций библиотеки SIGAR [4]. Реализация сторонних датчиков выполнена на специализированном языке, являющемся подмножеством языка ECMA Script [5] и поддерживающем вызовы внешних команд, обработку выходного потока, регулярные выражения и ряд других механизмов, позволяющих получать данные о состоянии нестандартных программно-аппаратных средств.

Агент промежуточного уровня устанавливается на промежуточных узлах РВС и выполняет функции сбора данных от группы агентов нижних уровней, агрегации и анализа этих данных, генерации и исполнения необходимых управляющих воздействий (в соответствии с правилами вывода, определенными для агента данного уровня), а также при необходимости осуществляет отправку данных агентам верхних уровней. Основное назначение агента промежуточного уровня – снижение нагрузки на агенты верхних уровней.

Агент верхнего уровня устанавливается в центральном узле РВС и выполняет

функции ядра системы, владеет полной информацией о состоянии улов РВС и предоставляет средства доступа к данным системы метамониторинга. Агент верхнего уровня может взаимодействовать с агентами промежуточного уровня, с агентами нижнего уровня, а также напрямую получать данные от локальных систем мониторинга и отдельных сенсоров. Собираемая информация накапливается в базе данных, унифицируется, агрегируется и предоставляется для доступа экспертной подсистеме и подсистемам уровня доступа. Экспертная подсистема агента верхнего уровня анализирует общее состояние РВС, генерирует управляющие воздействия, которые далее исполняются управляющей подсистемой. В базе знаний экспертной системы содержится информация о том, какие управляющие воздействия должны приниматься в автоматическом режиме, а какие при участии оператора РВС.

Экспертная подсистема. Для реализации функций экспертной подсистемы использована популярная среда CLIPS [6], основанная на продукционной модели знаний и предоставляющая программный интерфейс для работы с основными элементами экспертной подсистемы: базой знаний, содержащей списки фактов, списки объектов и правила вывода; механизмом вывода, реализующим логический вывод на основе сформированной базы знаний.

Для анализа значений метрик выбрана классификация по степени критичности, в соответствии с которой для каждой метрики определено четыре состояния: ошибка, критическое, предупреждение, удовлетворительное.

Значения метрик обрабатываются набором правил, которые оценивают эти значения в соответствии с введенными граничными параметрами и присваивают каждой метрике флаг состояния. Если в списке фактов появляется хотя бы одно состояние “ошибка” или “критическое”, то для избранных метрик проверяется история значений за определенный интервал, чтобы исключить явления, несущие случайный характер и неспособные в какой-либо мере повлиять на работу системы. Если информация о состоянии ошибки или критическом состоянии подтверждается, то генерируются управляющие воздействия, среди которых могут быть следующие: информирование агента верхнего уровня; создание и отправка сообщений оператору РВС; действия, направленные на автоматическое решение возникшей проблемы.

Подсистема накопления и обработки данных основана на принципах функционирования циклических баз данных (Round Robin Database) – баз данных, объем которых не меняется со временем. Фиксированный размер таких баз достигается благодаря заранее определенному количеству записей, используемых циклически для сохранения данных.

Несмотря на неоспоримые преимущества циклических баз данных, наиболее известные реализации (MRTG, RRDtools и др.) имеют недостатки, связанные, прежде всего, с низкой производительностью операций чтения/записи данных.

Проведенные эксперименты по созданию циклической базы данных на основе легковесной встраиваемой реляционной базы данных SQLite выявили еще более худшую производительность таких баз данных в сравнении с RRDtools и MRTG. В результате было принято решение о создании собственной реализации циклической базы данных, использующей для хранения структурированной информации специализированный текстовый формат, основанный на расширяемом метаязыке XML. Были реализованы механизмы чтения/записи данных, агрегации данных за определенный интервал времени, вытесне-

ния устаревших данных, выборки данных по определенным критериям, а также механизмы кеширования данных. Собственная реализация циклической базы данных показала высокую эффективность в сравнении с универсальными системами организации циклических баз данных (RRDtools, MRTG).

Для хранения правил вывода экспертной подсистемы, списка фактов, сгенерированных управляющих воздействий, шаблонов для трансляции управляющих воздействий в системные команды и других знаний используется легковесная встраиваемая реляционная база данных SQLite. Для выполнения перечисленных функций производительность SQLite является приемлемой, а возможность встраивания средств управления базой данных в состав программного агента позволяет сохранить малый размер программы и обеспечить минимальную зависимость от стороннего программного обеспечения, установленного на узле.

Графический пользовательский интерфейс является одним из наиболее важных компонентов любой системы, предполагающей взаимодействие с пользователем. Информативность и наглядность представления данных, актуальность отображаемой информации, возможность формирования комплексных отчетов по выбранным параметрам и ряд других возможностей делают систему полезной для различных категорий пользователей.

В качестве одного из существенных отличий предлагаемого подхода к построению интерфейса системы метамониторинга является реализация средств генерации графиков на клиентской стороне веб-приложения (в веб-браузере пользователя). Такой подход, во-первых, позволяет снизить нагрузку на центральный узел системы метамониторинга за счет переноса части нагрузки на компьютер пользователя. Во-вторых, при таком подходе появляется возможность отображать графики в режиме реального времени без периодической перезагрузки веб-страниц. В-третьих, в значительной мере снижается объем передаваемого трафика между клиентской и серверной сторонами веб-приложения.

В качестве языков разработки пользовательского веб-интерфейса были выбраны: язык программирования PHP для реализации серверной части веб-приложения и совокупность веб-технологий, языков программирования и библиотек (таких как HTML5, JavaScript, Ajax, ExtJS, jQuery) для реализации клиентской части веб-приложения. Пример отображения графиков в режиме реального времени с использованием графических средств библиотеки ExtJS приведен на рис. 2.

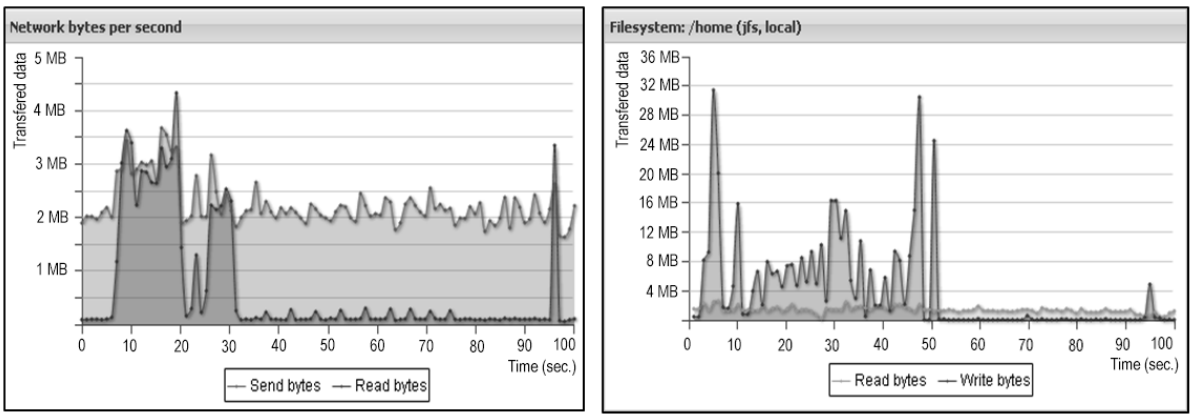


Рис. 2. Графики текущей загрузки сетевого интерфейса (слева) и файловой системы (справа)

На рис. 3 приведена схема размещения компонентов систем холодоснабжения и электроснабжения [7] вычислительного кластера «Академик В.М. Матросов» в машинном зале (слева) и на внешней площадке (справа) с отображением значений ключевых метрик. На рис. 4 представлен список значений метрик в табличной форме.

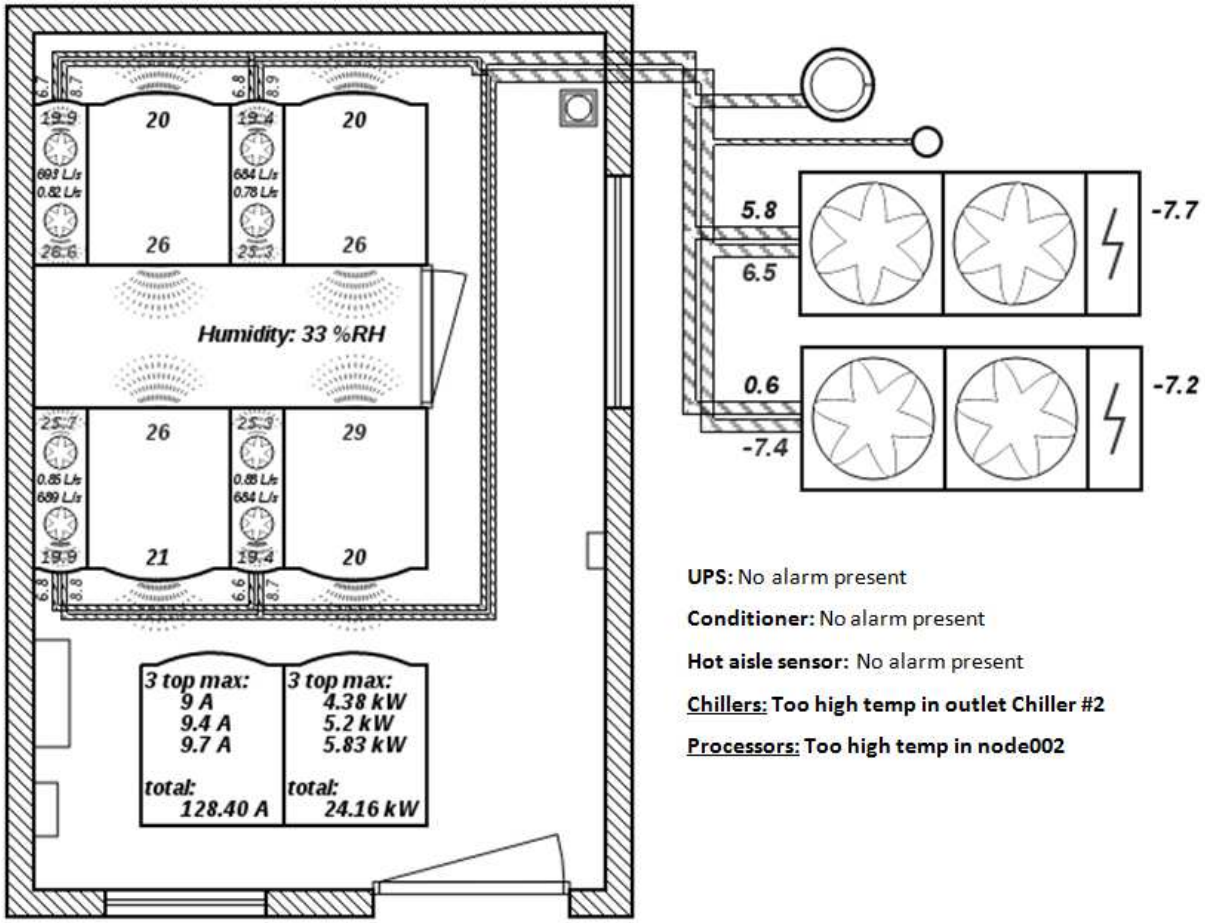


Рис. 3. Схема размещения компонентов систем холодоснабжения и электроснабжения с указанием значений ключевых метрик

Conditioners

Статус работы	On	On	On	On
Выходная мощность охлаждения	12.0 kW	10.7 kW	13.3 kW	9.4 kW
Необход. мощность охлаждения	12.0 kW	10.7 kW	13.3 kW	9.4 kW
Температура на входе	**21.3 °C**	**20.9 °C**	**21.3 °C**	**22.2 °C**
Темп. возвращаемого воздуха	20.1 °C	19.9 °C	20.0 °C	20.1 °C
Темп. приточного воздуха	**30.3 °C**	**30.9 °C**	**30.3 °C**	**29.8 °C**
Поток воздуха	**829 L/s**	**834 L/s**	**834 L/s**	**834 L/s**
Скорость вентиляторов	**53.4 %**	**53.9 %**	**53.9 %**	**53.9 %**
Состояние на входе	Open	Open	Open	Open
Состояние на выходе	Closed	Closed	Closed	Closed
Активный источник питания	A	A	A	A
Перепад давления на фильтре	0 Pa	5 Pa	17 Pa	15 Pa
Положение клапана в жидкости	**54 %**	**54 %**	**56 %**	**57 %**
Поток жидкости	**0.85 L/s**	**0.81 L/s**	**0.88 L/s**	**0.76 L/s**
Температура жидкости на входе	**6.2 °C**	**6.2 °C**	**5.9 °C**	**6.7 °C**
Температура жидкости на выходе	9.9 °C	9.6 °C	9.8 °C	9.9 °C
Общие установ. значения темп.	**22.2 °C**	**22.2 °C**	**22.2 °C**	**22.2 °C**
Общие установленные значения	20.0 °C	20.0 °C	20.0 °C	20.0 °C

UPS

#	Rating	Current	Power
1	16A	1.1 A	0.21 kW
	16A	0.0 A	0.00 kW
	16A	1.2 A	0.22 kW
2	16A	1.1 A	0.19 kW
	16A	0.0 A	0.00 kW
	16A	1.1 A	0.21 kW
3	32A	15.8 A	
	32A	9.4 A	8.38 kW
	32A	13.7 A	
4	32A	16.4 A	
	32A	18.9 A	10.47 kW
	32A	11.9 A	
5	**32A**	**15.8 A**	
	32A	**16.6 A**	**10.65 kW**
	32A	**15.7 A**	
6	**32A**	**15.3 A**	
	32A	**9.8 A**	**8.39 kW**
	32A	**13.3 A**	

Рис. 4. Список значений метрик в табличной форме

3. Вычислительный эксперимент

Разработанные методы и инструментальные средства метамониторинга РВС были успешно апробированы в суперкомпьютерном центре ИДСТУ СО РАН [8] при решении ресурсоемких задач в области биоинформатики, квантовой химии, логического поиска и других.

Экспериментальная РВС включала следующие программно-аппаратные ресурсы:

– 2 вычислительных узла с графическими ускорителями Nvidia Tesla C1060 с общим числом ядер 1920;
– 10 вычислительных узлов с процессорами AMD Opteron 6276 с общим числом ядер 320;
– 20 вычислительных узлов с процессорами Intel Xeon 5345 с общим числом ядер 160.

В процессе тестирования системы метамониторинга в РВС был выявлен перечень программно-аппаратных ресурсов, находившихся в критическом состоянии и в состоянии ошибки:

– (warning node node-4.blackford.icc.ru loadavg 43);
– (critical node node-13.blackford.icc.ru cpu-sys-p 77);
– (critical node node011.matrosov.icc.ru filesystem /store wtime 583816);
– (error node node-4.blackford.icc.ru available down).
– (warning node node-15.blackford.icc.ru memory-used-ten 81);
– (critical node node2.tesla.icc.ru memory-used-ten 92);
– (error node node-7.blackford.icc.ru filesystem /store du-free 0).

Анализ сведений о состоянии программно-аппаратных ресурсов РВС, собранных системой метамониторинга, позволил выявить неэффективную работу отдельных пользовательских приложений, произвести оптимизацию загрузки вычислительных ресурсов и повысить показатели надежности РВС.

Например, при обработке экспериментальных данных для аннотации генома байкальской диатомовой водоросли Synedra acus с использованием программного пакета MAKER [9] было выявлено преобладание операций чтения/записи в сетевую директорию по отношению к вычислительным операциям. Своевременная установка значений конфигурационных параметров пакета, указывающих на необходимость записи результатов вычислений в локальные директории узлов, позволила повысить эффективность работы пакета более чем на 30%.

Заключение

Новизна предложенного подхода к организации системы метамониторинга РВС заключается, в первую очередь, в создании универсальных программных агентов, способных вести сбор данных о состоянии узлов РВС, анализировать эти данные и принимать самостоятельные решения без участия оператора РВС. Мультиагентный подход позволил, во-первых, снизить общую нагрузку, создаваемую компонентами системы метамониторинга, во-вторых, повысить надежность РВС за счет децентрализованной генерации и исполнения управляющих воздействий. Универсальность архитектуры разработанного программного агента делает возможным его использование на различных уровнях иерархии системы метамониторинга, что, в свою очередь, обеспечивает высокую масштабируемость системы при ее использовании в РВС, состоящей из большого количества узлов.

Работа выполнена при частичной финансовой поддержке Совета по грантам Президента Российской Федерации для государственной поддержки ведущих научных школ (НШ-5007.2014.9).

Литература

1. Адинец, А.В. Мониторинг, анализ и визуализация потока заданий на кластерной системе / А.В. Адинец, П.А. Брызгалов, Вад.В. Воеводин // Высокопроизводительные параллельные вычисления на кластерных системах: Материалы XI Всерос. конф. – Нижний Новгород, 2011. – С. 10–14.
2. Сидоров, И.А. Один из подходов к реализации средств мониторинга высокопроизводительных вычислительных систем / И.А. Сидоров, В.В. Скоров // Информационные и математические технологии в науке и управлении: Тр. XVIII Байкальской Всерос. конф. – Ч. III. – Иркутск: Изд-во ИСЭМ СО РАН, 2013. – С. 177–184.
3. Wooldridge, M. Intelligent Agents: Theory and Practice / M. Wooldridge, N. Jennings // The Knowledge Engineering Review. – 1995. – Vol. 10, No 2. – P. 115–152.
4. System Information Gatherer and Reporter API. URL: http://www.hyperic.com/products/sigar (дата обращения: 08.04.2014).
5. Standard ECMA-262: ECMAScript Language Specification. URL: http://www.ecma-international.org/publications/standards/Ecma-262.htm (дата обращения: 08.04.2014)
6. Частиков, А.П. Разработка экспертных систем. Среда CLIPS / А. П. Частиков, Д. Л. Белов, Т. А. Гаврилова. – Санкт-Петербург: БХВ-Петербург, 2003. – 393 с.
7. Скоров В.В. Редактор графического представления объектов инженерной инфраструктуры вычислительного центра / В.В. Скоров, И.А. Сидоров, А.П. Новопашин // Ляпуновские чтения: Материалы конф. – Иркутск: Изд-во ИДСТУ СО РАН, 2013. – С. 53.
8. Иркутский суперкомпьютерный центр СО РАН. URL: http://hpc.icc.ru (дата обращения: 08.04.2014)
9. MAKER – genome annotation pipeline. URL: http://gmod.org/wiki/MAKER (дата обращения: 08.04.2014)

Сидоров Иван Александрович, к.т.н., н.с., Федеральное государственное бюджетное учреждение науки Институт динамики систем и теории управления СО РАН (Иркутск, Российская Федерация), ivan.sidorov@icc.ru

Новопашин Алексей Петрович, к.т.н., зав. лаб., Федеральное государственное бюджетное учреждение науки Институт динамики систем и теории управления СО РАН (Иркутск, Российская Федерация), apn@icc.ru

Опарин Геннадий Анатольевич, д.т.н., проф., зам. директора по научной работе, Федеральное государственное бюджетное учреждение науки Институт динамики систем и теории управления СО РАН (Иркутск, Российская Федерация), oparin@icc.ru

Скоров Владимир Владимирович, программист, Федеральное государственное бюджетное учреждение науки Институт динамики систем и теории управления СО РАН (Иркутск, Российская Федерация), skorov@icc.ru

Поступила в редакцию 10 апреля 2014 г.

Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 2, pp. 30–42

METHODS AND TOOLS OF DISTRIBUTED COMPUTING SYSTEMS METAMONITORING

I.A. Sidorov, Institute for System Dynamics and Control Theory of the Siberian Branch of the RAS (Irkutsk, Russian Federation)
A.P. Novopashin, Institute for System Dynamics and Control Theory of the Siberian Branch of the RAS (Irkutsk, Russian Federation)
G.A. Oparin, Institute for System Dynamics and Control Theory of the Siberian Branch of the RAS (Irkutsk, Russian Federation)
V.V. Skorov, Institute for System Dynamics and Control Theory of the Siberian Branch of the RAS (Irkutsk, Russian Federation)

This paper discusses the problems of monitoring a distributed computing environment that includes a lot of loosely-coupled heterogeneous computing resources. The principal distinctive feature and novelty aspect of the developed metamonitoring system first of all consists in creation of the universal software agents, capable to collect data about state of different environment components from local monitoring systems, to unify and analyze these data, and if necessary to generate and execute automatic control actions directed on changing equipment working mode, including the shutdown in case of emergencies. The approach is based on web-technologies and multi-agent technologies usage, methods of expert systems creation, methods of decentralized processing and the distributed storage of data.

Keywords: distributed computing systems, monitoring, multiagent technologies, expert systems

References

1. Adinets, A.V., Bryzgalov P.A., Voevodin V.V. Monitoring, analiz i visualizaciya potoka zadanii na klasternoi sisteme [Monitoring, analyze and visualization of tasks flow on cluster system]. Visokoproizvoditelnye parallelnye vichisleniya na klasternih sistemah [High-performance parallel computation on cluster systems].Nizhny Novgorod, 2011. P. 10-14.
2. Sidorov I.A., Skorov V.V. Odin iz podhodov k realizacii sredstv monitoringa visokoproizvoditelnih vichislitelnih system [Approach to the monitoring system implementation for HPC clusters] Informacionnye I matematicheskie technologii v nauke I upravlenii [Informational and mathematical technologies in science and control]. Irkutsk, Publishing of MESI SB RAS, 2013. P. 177-184.
3. Wooldridge M., Jennings N. Intelligent Agents: Theory and Practice // The Knowledge Engineering Review, 1995. Vol. 10, No 2. P. 115-152.
4. System Information Gatherer and Reporter API. URL: http://www.hyperic.com/products/sigar (accessed: 08.04.2014).
5. Standard ECMA-262: ECMAScript Language Specification. URL: http://www.ecma-international.org/publications/standards/Ecma-262.htm (дата обращения: 08.04.2014)
6. Chastikov A.P, Belov D.L, Gavrilova T.A. Razrabotka ekspertnyh sistem. Sreda CLIPS [Developing of expert systems. CLIPS Toolkit]. Saint-Petersburg, 2003. 393 P.

7. Skorov V.V., Sidorov I.A., Novopashin A.P Redaktor graficheskogo predstavleniya objectov injenernoi infrastructury vichislitelnogo tsentra [Editor of graphical representation of engineering infrastructure objects of HPC-center] // Lyapunovskie chteniya [Lyapunov reading]. Irkutsk, Publishing at ISDCT SB RAS, 2013. P. 53.
8. Irkutsk supercomputer center URL: http://hpc.icc.ru (accessed: 08.04.2014)
9. MAKER – genome annotation pipeline. URL: http://gmod.org/wiki/MAKER (accessed: 08.04.2014)

Received 10 April 2014.

УДК 004.75

МЕТОДЫ И ЭВРИСТИКИ ПЛАНИРОВАНИЯ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЯХ С НЕОТЧУЖДАЕМЫМИ РЕСУРСАМИ[1]

В.В. Топорков, А.В. Бобченков, Д.М. Емельянов, А.С. Целищев

Статья посвящена вопросам эффективной организации вычислений в виртуальных организациях распределенных сред с неотчуждаемыми ресурсами, включая коммерчески используемый грид. Предлагается и обосновывается комплексное сочетание методов и эвристик планирования и распределения ресурсов на уровне потоков заданий и приложений, реализующих принятую в виртуальной организации политику предоставления и потребления ресурсов с использованием экономических принципов.

Ключевые слова: распределенные вычисления, планирование, оптимизация, слот, альтернатива, циклическая схема планирования.

Введение

Решение больших задач в распределенных вычислительных средах требует привлечения значительных ресурсов, часть из которых может использоваться совместно с их владельцами [1–3]. Известные алгоритмы, их комбинации и эвристики, используемые в программах-планировщиках, как правило, не позволяют получать эффективные или близкие к оптимальным планы в условиях разнородных распределенных сред с динамично изменяющимся составом неотчуждаемых вычислительных ресурсов. В этих условиях весьма эффективными оказываются так называемые экономические модели планирования распределенных вычислений [1–3].

Среди различных подходов к планированию вычислений в распределенных средах можно выявить следующие тенденции. Одна из них основывается на использовании доступных ресурсов и планировании вычислений на уровне приложений. Зачастую при этом не предполагается наличия какого-либо регламента в предоставлении ресурсов [4]. Роль посредников между пользователями и вычислительными узлами выполняют агенты приложений – брокеры ресурсов. В рамках этого направления выполнен ряд проектов - AppLeS, APST, Legion, DRM, Condor-G, Nimrod/G и другие. Другая тенденция связана с образованием виртуальных организаций (ВО) пользователей и предполагает планирование на уровне потоков заданий. В этой модели внешний планировщик (грид-диспетчер, метапланировщик, метаброкер [5, 6]) выполняет роль промежуточного звена между пользователями и множеством систем управления заданиями на удаленных вычислительных узлах.

В рамках первого из направлений системы планирования и управления ресурсами являются хорошо масштабируемыми и адаптируемыми к особенностям приложений. Однако использование независимыми пользователями различных критериев для оптимизации планов выполнения своих заданий (в условиях возможной конкуренции с дру-

[1] Статья рекомендована к публикации программным комитетом Международной научной конференции «Параллельные вычислительные технологии – 2014».

гими заданиями) может ухудшать такие интегральные характеристики распределенной среды, как время выполнения пакета заданий и загрузка ресурсов. Образование ВО, с одной стороны, естественным образом ограничивает масштабируемость систем управления заданиями. С другой стороны, наличие определенных правил предоставления и потребления ресурсов, основанных, в частности, на экономических моделях [1–3, 5–9], позволяет повысить эффективность планирования и распределения ресурсов на уровне потоков заданий.

Идея «конвергенции» вышеупомянутых подходов была декларирована довольно давно [5, 10, 11]. Тем не менее, успешные попытки ее реализации обладают рядом серьезных ограничений. Так, в одних из известных моделей отыскивается лишь подходящий набор ресурсов [12–14] и не поддерживаются оптимизационные механизмы планирования заданий. В других моделях [5, 7, 8], не представлены аспекты, связанные с динамикой изменения загрузки узлов, конкуренцией независимых пользователей, а также глобальных и локальных потоков заданий. Подход к планированию в ВО, предлагаемый в данной работе, принципиально отличается от известных решений тем, что строится на сочетании методов и эвристик планирования параллельных приложений и управления потоками независимых заданий [10, 11].

1. Модель справедливого разделения ресурсов

1.1. Циклическая схема и стратегии планирования

Модель для управления выполнением независимых заданий основывается на экономических принципах, а планирование заданий выполняется циклично на альтернативных наборах предварительно отобранных ресурсов [15, 16]. В отличие от известных, рассматриваемая модель предполагает группирование заданий в потоки и пакеты в соответствии с их свойствами и потребностями в ресурсах, а также цикличное планирование системы заданий на основе динамично обновляемых стратегий и ограничений [15]. План выполнения системы заданий оптимизируется по совокупности критериев в соответствии с политикой справедливого предоставления и потребления ресурсов, принятой в виртуальной организации.

Циклический подход к планированию рассматривается в рамках модели, в основу которой заложена иерархическая схема управления потоками заданий [15, 16]. Планирование потока заданий осуществляется по пакетам на основе расписаний выполнения заданий в локальных процессорных узлах – в соответствии с циклической схемой планирования (ЦСП) (рис. 1).

Наборы доступных слотов и стоимость C_j их использования, определяемая их владельцами, динамично определяются по данным, поступающим от локальных менеджеров ресурсов или систем пакетной обработки заданий. В каждом цикле планирования требуется решение двух задач. Во-первых, необходимо отобрать подходящие (по производительности, времени и стоимости) наборы слотов – альтернативы выполнения для каждого задания пакета на данном цикле планирования. Каждая альтернатива характеризуется суммарной стоимостью, временем выполнения, старта, завершения и другими показателями, например, уровнем энергопотребления. Во-вторых, требуется выбрать комбинацию альтернатив, оптимальную или эффективную с позиций прохождения всего пакета заданий в соответствии с принятой политикой ВО.

Решению вышеупомянутых задач для вектора критериев, формализующих политику ВО и формирующих стратегии планирования, посвящена работа [15]. Здесь, для иллюстрации основной идеи, рассмотрим задачу выбора оптимальной комбинации слотов (с одним критерием).

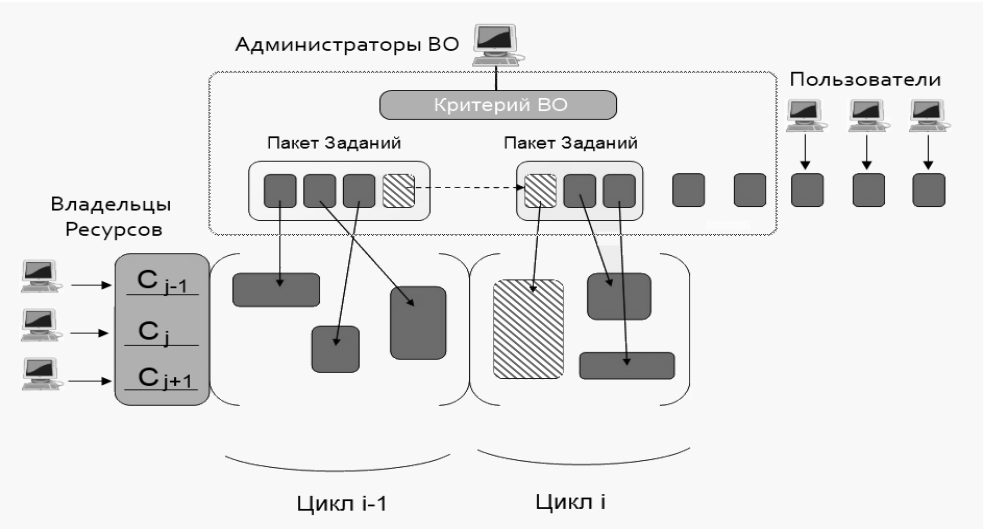


Рис. 1. Цикличное планирование потока заданий

Пусть S_i – множество наборов слотов, подходящих для выполнения i-го задания пакета, $i=1,...,n$. Набор слотов $s_j \in S_i$ является подходящим для i-го задания, если он удовлетворяет требованиям ресурсного запроса, цене $c_i(s_j)$ и времени $t_i(s_j)$, $j=1,...,N, N=\left|\bigcup_{i=1}^{n} S_i\right|$. При этом $S_1,...,S_n$ не обязательно не пересекаются. Комбинация представляет собой n-элементную последовательность $\bar{s}$ не обязательно различных наборов слотов. Любое из заданий может использовать один и только один набор слотов. В то же время некоторые из наборов могут разделяться по времени или ресурсу несколькими заданиями одного и того же пакета. Обозначим через S множество комбинаций слотов для пакета заданий. Если для какого-либо из заданий подходящего набора слотов не существует, то его планирование переносится в следующий цикл и задание занимает место в очереди цикла планирования в соответствии с некоторым приоритетом (см. рис. 1). После просмотра слотов для всех n заданий пакета отыскиваются альтернативные наборы слотов, поскольку при отборе слотов для i-го задания не весь исходный список может быть просмотрен. Эта процедура итеративно реализуется для всех заданий, планирование которых не перенесено в следующий цикл.

Пусть $f_i(s_j)$ – функция, определяющая эффективность выполнения i-го задания на наборе s_j слотов при допустимых затратах $g_i(s_j)$. В частности, $f_i(s_j)=c_i(s_j)$ – стоимость использования набора s_j в течение времени $g_i(s_j)=t_i(s_j)$ либо $f_i(s_j)=t_i(s_j)$ – время выполнения i-го задания на наборе s_j слотов стоимостью $g_i(s_j)=c_i(s_j)$. Полагаем, что критерий эффективности прохождения пакета заданий представляет собой сепарабельную функцию

$$f(\bar{s}) = \sum_{i=1}^{n} f_i(s_j),\ s_j \in S_i, j \in \{1,...,N\}, \bar{s} \in S. \quad (1)$$

Задается ограничение g^* на суммарные затраты для выполнения всего пакета заданий. В частности, для того чтобы не допустить монополизации использования какого-либо ресурса наиболее «богатыми» пользователями, вводится ограничение b^* на бюджет ВО – максимальное значение $c(\bar{s}) = \sum_{i=1}^{n} c_i(s_j), s_j \in S_i, j \in \{1,...,N\}, \bar{s} \in S$. Требование сбалансированности долей глобальных и локальных потоков заданий заставляет собственников вводить ограничение t^* на суммарное время занятия слотов. Ограничение g^* формируется динамично к началу текущего цикла планирования таким образом, что

$$g^* = \sum_{i=1}^{n} g_i^0(s_j), \quad (2)$$

где $g_i^0(s_j)$ – функция уровня ресурсных затрат, учитывающая характеристики подходящих наборов слотов для каждого задания пакета.

Формальная постановка задачи выбора оптимальной комбинации слотов $\bar{s}$ заключается в отыскании экстремума (1) при ограничении

$$\sum_{i=1}^{n} g_i(s_j) \le g^*, \quad (3)$$

где g^* определяется в соответствии с (2).

Пример функции $g_i^0(s_j)$ в (2) уровня временны́х затрат:

$$t_i^0(s_j) = [\sum_{s_j} t_i(s_j)/l_i], \quad (4)$$

где l_i – число подходящих (альтернативных) наборов слотов для i-го задания; $[\cdot]$ – ближайшее не большее целое число.

Тогда ограничение (3) с учетом (4) задается следующим образом:

$$t^* = \sum_{i=1}^{n} t_i^0(s_j). \quad (5)$$

Пакет состоит из n независимых заданий, что допускает естественную декомпозицию решения задачи (1)-(3) на n этапов с использованием методов динамического программирования. Для этого необходимо ввести понятие состояния системы (пакета заданий). Его будем представлять уровнем g_i допустимых суммарных затрат на выполнение части заданий пакета, включая i-е задание, например заданий $i, i+1,...,n$ либо заданий $i, i-1,...,1$. В частности, в одном случае g_i может представлять собой суммарное время t_i занятия слотов заданиями $i, i+1,...,n$ либо заданиями $i, i-1,...,1$. В другом случае, g_i – суммарная стоимость c_i использования слотов заданиями $i, i+1,...,n$ или $i, i-1,...,1$. Тогда рекуррентные соотношения для отыскания экстремума (1) при ограничении (3) и использовании набора слотов $s_j \in S_i$, $i = \overline{1,n}, j \in \{1,...,N\}$ по схеме обратной прогонки имеют следующий вид:

$$f_i(g_i) = \operatorname*{extr}_{s_j \in S_i} \{f_i(s_j) + f_{i+1}(g_i - g_i(s_j))\},\ g_i(s_j) \le g_i \le g^*,\ i = \overline{1,n},$$

$$f_{n+1}(g_{n+1}) \equiv 0,\ g_i = g_{i-1} - g_{i-1}(s_k),\ 1 < i \le n,\ g_1 = g^*,\ s_k \in S_{i-1}, \quad (6)$$

где g_i – суммарные затраты при использовании наборов слотов заданиями $i, i+1,...,n$ пакета.

Соответствующие уравнения для схемы прямой прогонки:

$$f_i(g_i) = \operatorname*{extr}_{s_j \in S_i} \left\{ f_i(s_j) + f_{i-1}(g_i - g_i(s_j)) \right\},\ g_i(s_j) \le g_i \le g^*,\ i = \overline{1,n},$$

$$f_0(g_0) \equiv 0,\ g_i = g_{i+1} - g_{i+1}(s_k),\ 1 \le i < n,\ g_n = g^*,\ s_k \in S_{i+1}, \tag{7}$$

где g_i – суммарные затраты на выполнение заданий $i, i-1,...,1$ пакета.

Оптимальные затраты в схемах (6), (7) определяются из уравнения

$$g_i^*(s_j) = \arg \operatorname*{extr}_{g_i(s_j) \le g_i} f_i(g_i),\ i = \overline{1,n}. \tag{8}$$

Оптимальный набор слотов для выполнения i-го задания будем обозначать через $s_i^* \in \{1,...,N\}$. В обеих схемах (6) и (7) он задается соотношением

$$s_i^* = \arg \operatorname*{extr}_{s_j \in S_i} f_i(g_i * (s_j)),\ i = \overline{1,n}. \tag{9}$$

На основе функциональных уравнений (6)-(9) отыскивается решение задачи (1)-(3), позволяющее получить оптимальную комбинацию слотов $(s_1^*,...,s_n^*)$ для пакета заданий.

Рассмотрим примеры стратегий планирования, имеющие важное практическое значение.

1. Максимизация доходов собственников ресурсов при заданном ограничении на суммарное время использования слотов в соответствии с (4), (5)

$$f_i(t_i) = \max_{s_j \in S_i} \left\{ c_i(s_j) + f_{i+1}(t_i - t_i(s_j)) \right\},\ i = 1,...,n,\ f_{n+1}(t_{n+1}) \equiv 0. \tag{10}$$

2. Минимизация суммарного времени завершения пакета заданий при ограничении на бюджет b^* виртуальной организации

$$f_i(c_i) = \min_{s_j \in S_i} \left\{ t_i(s_j) + f_{i+1}(c_i - c_i(s_j)) \right\},\ i = 1,...,n,\ f_{n+1}(c_{n+1}) \equiv 0. \tag{11}$$

3. Минимизация суммарной стоимости выполнения пакета заданий при ограничении на суммарное время (5) использования слотов

$$f_i(t_i) = \min_{s_j \in S_i} \left\{ c_i(s_j) + f_{i+1}(t_i - t_i(s_j)) \right\},\ i = 1,...,n,\ f_{n+1}(t_{n+1}) \equiv 0. \tag{12}$$

4. Минимизация простоя ресурсов при ограничении на время (5) их использования

$$f_i(t_i) = \max_{s_j \in S_i} \left\{ t_i(s_j) + f_{i+1}(t_i - t_i(s_j)) \right\},\ i = 1,...,n,\ f_{n+1}(t_{n+1}) \equiv 0. \tag{13}$$

Вышеприведенные функциональные уравнения (10)–(13) представляют собой конкретизацию (6) и реализуются в виде компонентов симулятора распределенной вычислительной среды [16].

ЦСП присущи следующие ограничения.

Отсутствует возможность влиять на ход выполнения отдельного пользовательского задания: поиск отдельных альтернатив осуществляется по принципу First Fit, а выбор их оптимальной комбинации согласно (9) отражает интересы всей ВО. Таким образом, не учитываются предпочтения отдельных пользователей, что не позволяет говорить о справедливом разделении ресурсов.

Планирование выполнения пакета заданий основывается на пользовательской, зачастую весьма неточной, оценке времени $t_i(s_j)$ выполнения конкретного приложения. В случае несостоятельности оценки преждевременно высвобожденные ресурсы могут про-

стаивать, тем самым, понижая уровень загрузки среды. Для планирования пакета заданий необходимо выделение нескольких «непересекающихся» (по занятым ресурсам и времени) альтернатив, а для выполнения задания выбирается одна альтернатива. Это может приводить к фрагментации ресурсов.

Известный алгоритм обратного заполнения - бэкфиллинг [17], используемый, как правило, в параллельных системах, в ряде случаев позволяет ее предотвратить. Бэкфиллинг является модификацией метода FCFS и требует от пользователей оценки времени выполнения их заданий, что позволяет заранее, на этапе планирования резервировать ресурсы. С помощью механизма предварительного резервирования метод гарантирует получение ресурсов высокоприоритетными заданиями, и, вместе с тем, допускает нарушение порядка очереди, что способствует повышению коэффициента общей загрузки ресурсов. Нарушение порядка очереди происходит на этапе обратного заполнения, при попытке назначить на доступные ресурсы задания с меньшим приоритетом. При использовании консервативной вариации задание с меньшим приоритетом может быть запущено вне очереди только в том случае, если оно не будет препятствовать запуску всех более приоритетных заданий. Агрессивный бэкфиллинг позволяет запускать задания вне очереди лишь в том случае, если они не будут препятствовать запуску самого приоритетного задания. Бэкфиллинг имеет такие недостатки, как неэффективное использование ресурсов по критериям, отличным от времени старта (особенно при относительно невысоком уровне загрузки доступных ресурсов), а также невозможность принципиально повлиять на качество распределения, изменяя политики и критерии, действующие в рамках ВО. Тем не менее, целесообразно рассмотреть возможность использования бэкфиллинга в сочетании с ЦСП и различными стратегиями для предотвращения фрагментации ресурсов.

1.2. Требования к модели справедливого разделения ресурсов и модификация ЦСП

Сформулируем требования к модели планирования и справедливого разделения ресурсов (МПСР) среди участников ВО путем обеспечения следующих дополнительных возможностей.

Для администраторов ВО - управления процессом планирования за счет определения политики прохождения потока заданий. Для пользователей - влияния на план выполнения заданий путем введения критерия эффективности. Для владельцев - управления загрузкой собственных ресурсов установлением удельной стоимости их использования в течение интервала планирования.

МПСР представляет собой развитие и обобщение ЦСП. Для учета интересов пользователей в формат ресурсного запроса дополнительно вводится критерий оптимизации. В отличие от так называемых мягких ограничений [14], представляющих пользовательские предпочтения, критерий оптимизации, заданный в ресурсном запросе, в дальнейшем учитывается на этапе поиска альтернатив выполнения. Для выбора альтернатив, оптимальных по заданному критерию, используется алгоритм AEP (Algorithm searching for Extreme Performance), подробно описанный в [18]. Таким образом, формирование набора альтернатив выполнения осуществляется с учетом интересов пользователей, согласно заданным критериям оптимизации отдельных заданий, а выбор оптимальной комбинации альтернатив осуществляется по критерию (1), формализующему соответ-

ствующий аспект политики ВО. Владельцы ресурсов получают возможность управлять прибылью и загрузкой вычислительных узлов за счет варьирования расписания доступности и запрашиваемой стоимости использования на интервале планирования.

Другое отличие МПСР от ЦСП состоит в алгоритме формирования системы заданий. МПСР предполагает разделение исходного пакета заданий на множество подпакетов и планирование каждого подпакета в отдельности на заданном интервале планирования. Необходимость процедуры «разрезания» пакета в МПСР особенно ярко проявляется при высоком уровне загрузки ресурсов среды. Согласно реализуемому в ЦСП механизму поиска множества альтернатив [16], при планировании пакета число найденных альтернатив может быть относительно невысоким (вплоть до единственной альтернативы на задание при высоком уровне загрузки доступных ресурсов). Это отражается на этапе выбора лучшей комбинации слотов (9) и на эффективности планирования всего пакета.

«Разрезание» пакета позволяет увеличить количество альтернатив для более приоритетных заданий, разнообразить по параметрам множество доступных альтернатив на этапе выбора оптимальной комбинации и, тем самым, повысить эффективность разделения ресурсов. При разбиении исходного пакета на n подпакетов, где n – общее количество заданий в пакете, алгоритм согласно приоритетам позволит найти лучшие наборы слотов для выполнения каждого из заданий в отдельности, не учитывая эффективность планирования пакета в целом. При «выделении» из исходного пакета лишь одного подпакета, результат будет идентичен применению ЦСП (см. рис. 1).

1.3. Комбинация циклического планирования и бэкфиллинга

МПСР позволяет оптимизировать выполнение потока заданий согласно интересам участников ВО при условии, что найдены альтернативные наборы слотов для выполнения заданий. Бэкфиллинг реагирует на досрочное освобождение ресурсов и допускает перепланирование «на лету», что очень важно в условиях, когда оценка времени выполнения и действительное время выполнения задания могут существенно отличаться.

Предлагается комбинированный, эвристический подход. В каждом цикле планирования из исходного пакета выделяется подмножество приоритетных заданий, например, самых «дорогих» (по стоимости выполнения), либо самых требовательных к ресурсу (по времени выполнения). Эти задания группируются в отдельный подпакет и планируются, возможно, и без соблюдения дисциплины очереди. Далее планирование этого подпакета производится с помощью МПСР на основе актуального расписания загрузки среды на рассматриваемом интервале планирования. Планирование остальных, менее требовательных к ресурсам заданий осуществляется с помощью бэкфиллинга на основе динамично обновляющейся информации о реальной загрузке узлов. Таким образом, метод планирования с использованием бэкфиллинга (МПБ) совмещает в себе преимущества как МПСР, так и бэкфиллинга, а именно, оптимизируется выполнение наиболее трудоемких заданий, эффективно используются ресурсы, преимущественно соблюдается дисциплина очереди выполнения заданий и достигаются невысокие значения времени старта.

1.4. Эвристическая процедура сдвига

Следующий шаг в развитии ЦСП – применение эвристической процедуры сдвига (ЦСПС) альтернатив выполнения задания к началу интервала планирования с сохране-

нием назначения на ресурсы. Суть процедуры сдвига (ПС) состоит в следующем. При поиске множества альтернатив выполнения и последующем выборе их оптимальной комбинации, выбранные альтернативы могут располагаться на рассматриваемом интервале планирования произвольным образом (на рис. 2а выбранные для трех заданий альтернативы схематически представлены серыми прямоугольниками в плоскости «Ресурсы – Время»). Из-за этой особенности не только увеличивается среднее время старта заданий пакета (по сравнению, например, с бэкфиллингом), но, также, происходит «дробление» слотов, образование автономных участков, в которых может быть затруднительно выделить необходимое для выполнения очередного задания количество ресурсов.

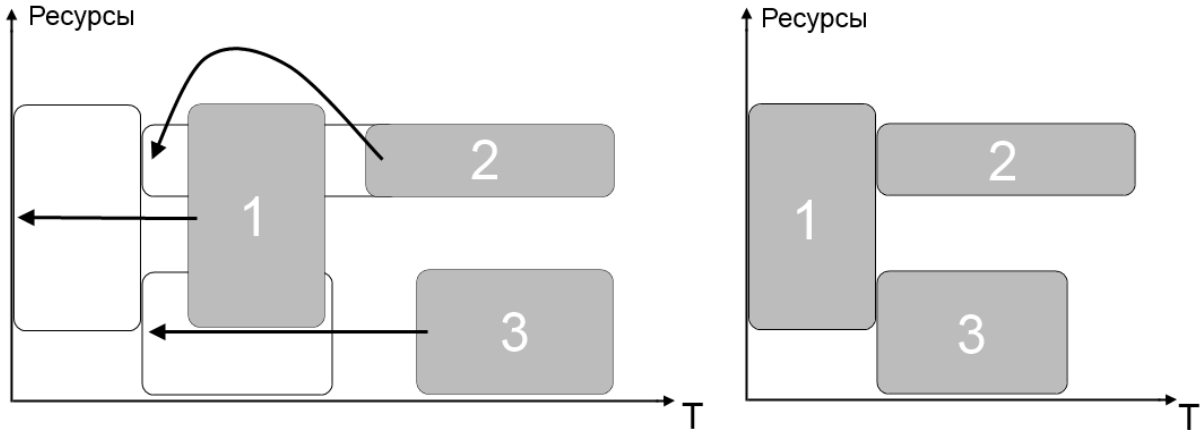


Рис. 2. Демонстрация процедуры сдвига: а) слева – исходное расположение альтернатив выполнения; б) справа – расположение альтернатив после применения процедуры сдвига

Процедура сдвига выполняется итеративно, последовательно для каждого задания. Очередность выборки заданий формируется согласно времени старта выбранных альтернатив: в первую очередь происходит попытка смещения альтернатив, имеющих наименьшее время старта. Такая очередность гарантирует, что при смещении любого задания, все более ранние задания уже будут смещены, и, таким образом, не будут занимать соответствующие узлы. Иначе возможна ситуация, что задание с меньшим временем старта и меньшим приоритетом заблокирует перемещение задания с более высоким приоритетом, а затем, в свою очередь будет смещено, освободив дополнительные слоты.

Попытка смещения заключается в проверке возможности выполнения задания на узлах, на которых уже было запланировано выполнение, в более ранний момент времени. В случае если узлы свободны – план выполнения задания переназначается на более раннее время старта (рис. 2б). При этом сохраняются все основные характеристики альтернативы, такие, например, как стоимость или общее время выполнения.

Применение процедуры сдвига позволит существенно уменьшить среднее время старта пакета заданий, планирование которого осуществлялось ЦСП или ее модификациями в условиях избыточного количество доступных ресурсов, за счет более плотного расположения заданий в начале интервала планирования.

2. Исследование подходов к справедливому разделению ресурсов

2.1. Постановка эксперимента

Для исследования предложенных подходов, а также сравнения с известными алгоритмами планирования использован симулятор распределенной вычислительной среды [16], дополненный компонентами поддержки МПСР. Для постановки экспериментов используются следующие общие настройки компонентов модели (в относительных единицах симулятора). Типовой размер цикла планирования принят равным 600 единицам времени. Количество узлов в ресурсном домене равно 24. Производительность узлов является случайной равномерно распределенной величиной $p \in [2, 10]$. Таким образом, самые производительные ресурсы ($p = 10$) смогут выполнять задания примерно в два раза быстрее средних по производительности ($p = 6$), а наименее производительные ($p = 2$) – в три раза медленнее. При этом обеспечивается достаточный уровень разнообразия ресурсов, отличие наименее и наиболее производительных экземпляров не будет превышать одного порядка внутри конкретного домена. Равномерный закон распределения выбран из предположения, что в рассматриваемой системе планирования состав доменов формируется с помощью искусственного отбора ресурсов (фильтрации) по таким характеристикам, как тип, производительность, расположение и т.п. (жесткие ограничения в [5]). Удельная стоимость ресурса назначается на этапе ценообразования в зависимости от производительности и случайной величины «скидка/наценка», подчиняющейся нормальному распределению. Количество заданий на каждом цикле планирования равно 20. Максимальная стоимость, которую пользователь готов заплатить за выполнение задания, задается таким образом, чтобы «богатейшие» пользователи могли позволить себе использование «дорогих» ресурсов (рыночная стоимость + наценка в 60%), а «беднейшие» были вынуждены рассчитывать на скидки. Эти факторы предотвращают монополизацию наиболее производительных и, соответственно, самых дорогостоящих ресурсов.

2.2. Учет интересов пользователей

Учет интересов пользователей и анализ результатов планирования для отдельных заданий происходит на основе сравнения МПСР и исходной ЦСП. Планирование пакета заданий в рамках МПСР, поиск альтернатив в которой осуществляется по алгоритму AEP [18], выполняется с учетом критерия, указанного в ресурсном запросе задания. Планирование пакета заданий по ЦСП предполагает выбор первой подходящей альтернативы без оптимизации.

Таблица 1 демонстрирует результаты планирования отдельных заданий в зависимости от заданного пользователем критерия оптимизации: времени старта и завершения задания, времени и стоимости выполнения задания (AEP минимизирует значение заданного критерия). Выбор одного из четырех критериев происходит на этапе генерации пакета заданий случайно по равномерному закону распределения. Равномерный закон распределения выбран из соображения того, что невозможно выделить один критерий оптимизации для независимых пользователей. В последней строке табл. 1 приведены результаты планирования того же пакета заданий с помощью ЦСП. Проведено моделирование 5000 отдельных циклов планирования. Наилучшие значения по критериям вре-

мени старта и завершения заданий, а также по времени и стоимости выполнения имеют задания, для которых был задан соответствующий критерий оптимизации (столбец «Критерий»). Исключение составляет подход с минимизацией времени завершения. Задания, для которых был указан соответствующий критерий оптимизации, в среднем обеспечивают не только наименьшее время старта, но и наименьшее время завершения. В целом, использование критерия оптимизации в МПСР по сравнению с ЦСП при выполнении отдельных заданий позволяет более чем на 23% уменьшить время старта завершения заданий, на 21% уменьшить общее время выполнения и на 12% снизить стоимость выполнения. Среднее количество N_A альтернатив выполнения в табл. 1, найденных для заданий в одном цикле планирования, практически не зависит от критерия оптимизации. Среднее число заданий на каждую группу, имеющую один и тот же критерий оптимизации, в среднем равно 5, что согласуется с использованием равномерного распределения при выборе одного из четырех рассматриваемых критериев оптимизации для каждого из 20 заданий пакета.

Таблица 1

Учет интересов пользователей ВО.

Критерий	N_A	**Время старта**	**Время выполнения**	**Время завершения**	**Стоимость**
Время старта	12,8	171,7	56,1	227,8	1281,1
Время выполнения	10,6	214,5	**39,3**	253,9	1278,5
Время завершения	12,2	**169,6**	45	**205,5**	1283,2
Стоимость	12,9	262,6	55,5	318	**1098,3**
ЦСП	12,1	222	50,3	272,3	1248,4

Из анализа результатов планирования отдельных заданий следует, что в рамках МПСР пользователи получают возможность влиять на эффективность выполнения собственных приложений путем введения соответствующего критерия оптимизации. Достигается это за счет того, что при наличии различных требований к эффективности выполнения заданий, ресурсы распределяются между заданиями неравномерно: в зависимости от используемого при отборе критерия. Заметим, что в исходной ЦСП на этапе поиска альтернатив все ресурсы распределяются между заданиями пакета равномерно.

2.3. Оптимизация выполнения пакета заданий

Следующий эксперимент посвящен сравнительному анализу результатов планирования при делении исходного пакета заданий в МПСР на подпакеты при различных уровнях загрузки среды. При выборе оптимальной комбинации альтернатив выполнения решалась задача минимизации среднего процессорного времени выполнения заданий $T_{проц}$. Процессорное время для альтернативы вычисляется как сумма длительностей слотов, входящих в сформированное «окно».

На рис. 3 представлены значения $T_{проц}$ в зависимости от числа $k \in \{1,2,3,5,6,10,20\}$ подпакетов, на которое делится исходный пакет заданий и уровня загрузки среды. Уровень загрузки при проведении серии экспериментов определяется средним относительным числом Y неудач – циклов планирования, в ходе которых не удается найти план выполнения для всех заданий пакета. Эксперименты проводились при высоком ($Y = 0.3$), среднем ($Y = 0.03$) и низком уровнях загрузки ($Y < 0.0002$) (на рис. 3 – ВЗС, СЗС и НЗС соответственно). Таким образом, количество неудач в проводимых сериях экспериментов отличается как минимум на порядок.

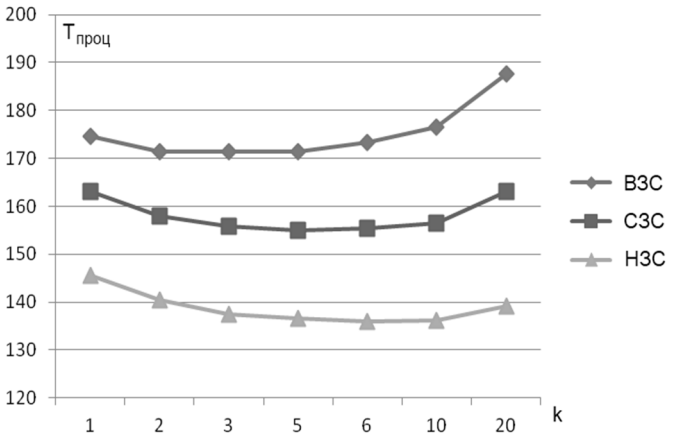


Рис. 3. Среднее время $T_{проц}$ выполнения заданий пакета в зависимости от количества k подпакетов

Анализ результатов показывает, что увеличение числа формируемых подпакетов обуславливает рост числа альтернатив для выполнения отдельного задания, уменьшение суммарной стоимости прохождения пакета заданий и увеличение относительного числа неудач Y. При увеличении уровня доступных ресурсов увеличивается число альтернатив для выполнения отдельного задания, уменьшается относительное количество неудач Y и уменьшается стоимость прохождения пакета заданий. Результаты экспериментов подтверждают, что разделение исходного пакета на подпакеты и их последовательное независимое планирование позволяет увеличить число альтернатив выполнения для заданий пакета и позволяет построить более эффективные планы. Так, при высокой загрузке вычислительной среды, наибольшую эффективность и наименьшее количество неудач обеспечивает разделение на меньшее количество подпакетов. С другой стороны, при низкой загрузке доступных ресурсов, целесообразно деление на большее количество подпакетов вплоть до планирования заданий по отдельности (см. рис. 3).

Другое преимущество МПСР, по сравнению с ЦСП, это - уменьшение стоимости прохождения пакета заданий при уменьшении уровня загрузки ресурсов. ЦСП стремится использовать весь доступный для выполнения пакета заданий бюджет $b*$, выбирая соответствующую комбинацию альтернатив. В то же время при планировании подпакетов с небольшим числом заданий, выбор зачастую ограничен несколькими альтернативами, стоимость которых не обязательно будет близка к установленному ограничению на бюджет.

Таким образом, МПСР позволяет не только учитывать интересы администраторов ВО (за счет оптимизации на этапе выбора комбинации альтернатив, как и в ЦСП), но и

за счет разбиения на подпакеты способна обеспечивать лучшее значение целевого критерия по сравнению с исходной схемой.

2.4. Учет интересов владельцев

Таблица 2 демонстрирует результаты планирования в МПСР с точки зрения владельцев ресурсов на примере характеристик отдельного вычислительного узла в зависимости от удельной стоимости c, назначенной на интервале планирования $T = 600$. В табл. 2: L_c –суммарное процессорное время загрузки узла на интервале планирования; U – средняя относительная величина загрузки ресурса на рассматриваемом интервале планирования, P – средняя прибыль, полученная владельцем ресурса, Y – относительное количество неудачных попыток планирования.

Таблица 2

Учет интересов владельцев ресурсов ВО

c	L_c	U	P	Y
2	256,6	0,44	527,1	0
4	234,9	0,39	939,6	0,001
6	185,4	0,31	**1112,3**	0,013
8	109,8	0,18	878,7	0,024
10	71	0,12	710,3	0,025

Из табл. 2 видно, что владельцы ресурсов в МПСР могут управлять собственной прибылью P и уровнем загрузки U вычислительных узлов на интервале планирования T путем выставления удельной стоимости c использования узла. Экстремум прибыли достигается при выставлении стоимости, близкой к «среднерыночной», то есть средней стоимости за экземпляр ресурса аналогичной производительности, выставленной другими владельцами ресурсов.

3. Эффективность использования ресурсов

3.1. Комбинированный метод планирования

Эффективность планирования с помощью комбинированного подхода МПБ может быть рассмотрена одновременно с позиций: значения критерия оптимизации в конкретной ВО, например, минимизации общего времени выполнения пакета заданий (11), и уровня загрузки среды, а также минимизации времени старта заданий пакета.

На рис. 4 представлены среднее время выполнения $T_{проц}$ и среднее время старта $T_{старт}$ заданий пакета в зависимости от соотношения $N_{цсп}/N$, в котором происходит разделение N заданий на подпакеты, $N_{цсп}$ – число заданий в первом подпакете, планирование которых осуществляется на основе ЦСП. При разделении на подпакеты используется простой приоритет – очередность заданий в пакете, характеристики самих заданий не принимаются во внимание. Представленные на рис. 4 результаты получены на основе моделирования 5000 независимых циклов планирования при среднем уровне загрузки ресурсов согласно настройкам, описанным в разделе 3. Из рис. 4 видно, что если планирование большей части задания осуществлять с помощью МПСР, достигается

лучшее значения целевого критерия ВО – времени выполнения $T_{проц}$, при этом среднее время старта $T_{старт}$ заданий отодвигается. И наоборот, если планирование большей части заданий осуществлять с помощью бэкфиллинга, среднее время старта приближается к началу интервала планирования, но значение целевого критерия оптимизации ухудшается. Особое внимание стоит уделить точке пересечения рассматриваемых графиков $T_{проц}$ и $T_{старт}$ на рис. 4. Ее наличие при монотонности графиков позволяет говорить о возможности достижения компромисса между средним временем старта заданий и значением целевого критерия в ВО.

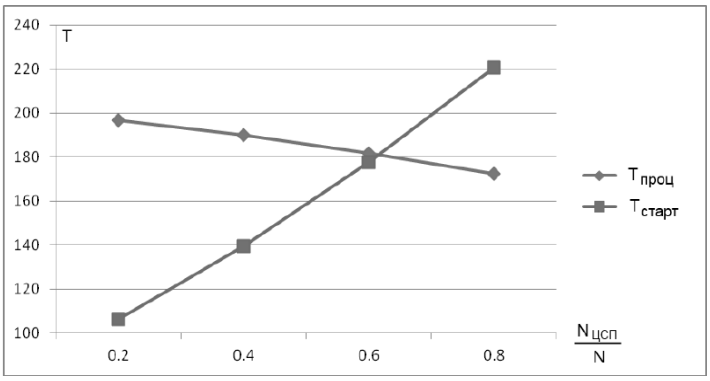


Рис. 4. Среднее время выполнения $T_{проц}$ и старта $T_{старт}$ заданий в МПБ

Таким образом, эвристика МПБ демонстрирует «компромиссные» значения рассматриваемых показателей по сравнению с МПСР и бэкфиллингом.

3.2. Состоятельность планов

МПСР и ЦСП позволяют строить предварительные планы выполнения пакета заданий на рассматриваемом интервале планирования, но не учитывают ситуаций, когда фактическое время выполнения заданий может оказаться меньше заявленного пользователем. Раннее завершение заданий и преждевременное освобождение и простой ресурсов могут негативно сказаться на эффективности выполнения пакета заданий по критериям, определенным участниками ВО, и сделать построенный план несостоятельным. С другой стороны, бэкфиллинг осуществляет планирование на основе динамически обновляемой информации о статусе выполнения заданий и загрузке вычислительных узлов и за счет этого может обеспечивать более эффективное прохождение потока заданий.

Для исследования и сравнения эффективности планов, построенных ЦСП, МПСР и бэкфиллингом было проведено имитационное моделирование, в котором реальное время выполнения заданий существенно отличается от времени предварительного резервирования ресурсов. Фактическое время выполнения заданий задавалось случайной величиной, подчиняющейся равномерному распределению на интервале $[0.2{*}T_{рез},\ T_{рез}]$, где $T_{рез}$ – время резервирования ресурсов, отведенное для выполнения задания. Равномерное распределение выбрано потому, что практически невозможно прогнозировать реальное время выполнения задания на выбранных ресурсах.

Таблица 3 содержит значения среднего времени выполнения (критерия оптимизации) и среднего времени старта заданий, полученные на этапе предварительного планирования на основе оценки времени выполнения заданий $T_{рез}$ (колонка «Запланированное») и в результате моделирования выполнения составленного плана с учетом фактического времени выполнения заданий (колонка «Фактическое»). Из табл. 3 видно, что даже при значительной разнице между временем резервирования ресурсов и фактическим временем выполнения заданий преимущество МПСР перед бэкфиллингом по целевому критерию оптимизации в ВО не только сохраняется, но и увеличивается. Объясняется это тем, что бэкфиллинг не осуществляет оптимизации по критериям, отличным от времени старта, а более плотное расположение заданий (реальное время старта заданий, планирование которых осуществляется с помощью бэкфиллинга, сокращается в среднем на 46%) задействует практически все доступные ресурсы, в том числе менее предпочтительные по целевому критерию. Таким образом, результаты эксперимента показывают, что предварительно составленные планы являются состоятельными по критериям, заданным в ВО, в случае, когда фактическое время выполнения существенно отличается от времени резервирования ресурсов.

Таблица 3

Фактическое и запланированное время выполнения заданий

Алгоритм	**Время выполнения**		**Среднее время старта заданий**	
	Запланированное	Фактическое	Запланированное	Фактическое
Бэкфиллинг	187,7	115,1	**69**	**37,3**
ЦСП	150,1	90,4	281,2	281,2
МПСР	**138,6**	**83,5**	223,8	223,8
Преимущество МПСР перед бэкфиллингом	26,2%	27,5%	-69%	-83%

Следует отметить, что дополнительное преимущество может быть достигнуто за счет учета информации о текущей загрузке вычислительных узлов и перепланирования.

3.3. Влияние числа узлов на эффективность планирования

Эффективность планирования в зависимости от числа узлов вычислительной среды оценивается по следующим критериям: общее процессорное время $T_{проц}$, среднее время старта $T_{старт}$ и завершения выполнения $T_{зав}$ пакета заданий, а также комбинированный показатель $F = T_{старт} + T_{проц}$. Проведен сравнительный анализ результатов планирования с применением ЦСП и ЦСП с эвристикой сдвига, МПБ и бэкфиллинга.

На рис. 5 представлено среднее время старта заданий для рассматриваемых подходов в зависимости от количества узлов вычислительной среды. При увеличении количества доступных узлов бэкфиллинг позволяет уменьшить среднее время старта вплоть до нуля (то есть все задания стартуют с начала интервала планирования). При этом время старта заданий, при использовании ЦСП практически не зависит от уровня доступных

ресурсов: оптимальный набор альтернатив находится примерно в середине интервала планирования. МПБ ожидаемо позволяет обеспечить время старта близкое к бэкфиллингу за счет заполнения неиспользованных циклической схемой слотов наименее приоритетными заданиями.

При большом количестве ресурсов время старта заданий в схеме ЦСПС стремится к ненулевому значению, так как при выборе оптимального набора альтернатив наиболее эффективные по критерию оптимизации ресурсы, как правило, используются более, чем одним заданием. Таким образом, в условиях разнородных ресурсов обеспечить старт всех заданий с самого начала интервала планирования практически невозможно.

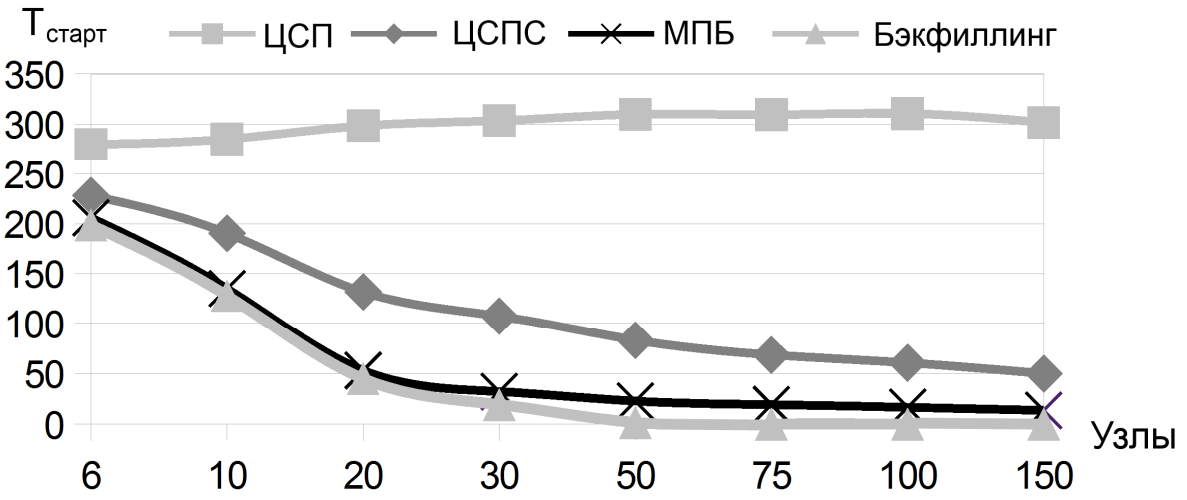


Рис. 5. Среднее время старта заданий пакета

Рис. 6 демонстрирует преимущество ЦСП и МПБ перед бэкфиллингом по целевому критерию оптимизации в ВО – процессорному времени выполнения заданий $T_{проц}$. Следует отметить, что с увеличением числа узлов преимущество ЦСП, ЦСПС и МПБ над бэкфиллингом становится больше. Применение дополнительной эвристики разделения на подпакеты может обеспечить еще большее преимущество циклической схемы по данному критерию.

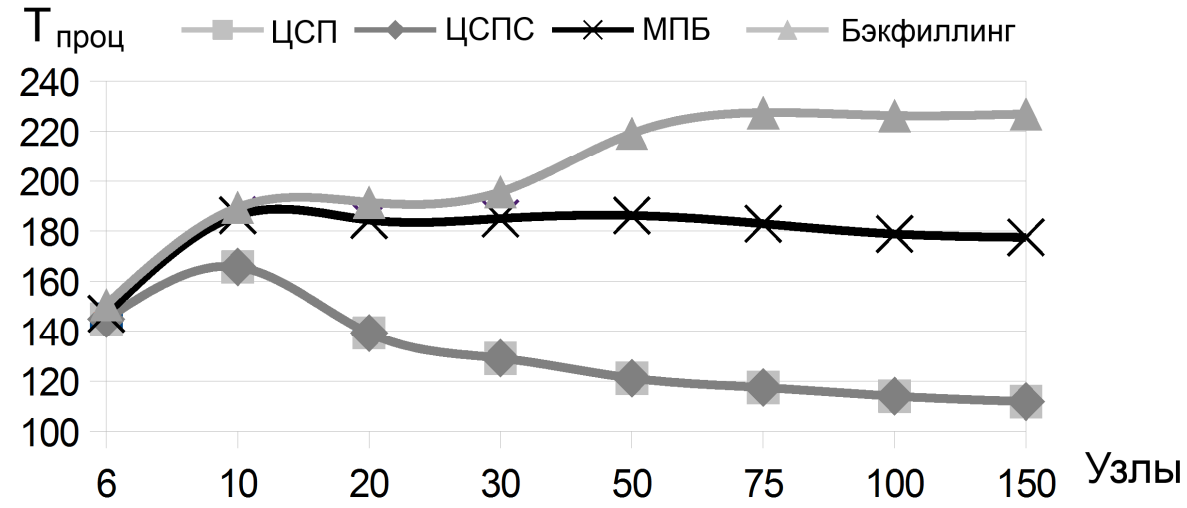


Рис. 6. Среднее процессорное время выполнения заданий пакета

На рис. 7 показана зависимость комбинированного показателя $F = T_{старт} + T_{проц}$ эффективности использования ресурсов от числа узлов.

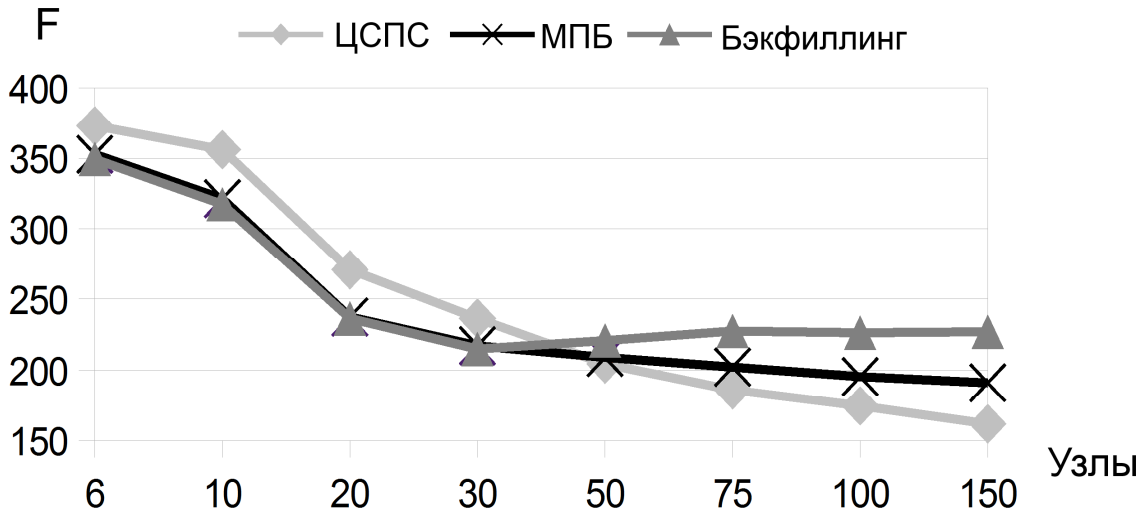


Рис. 7. Среднее значение F для заданий пакета

Важно отметить, что из пересечения графиков зависимостей для ЦСПС, МПБ и бэкфиллинга следует, что при относительно небольшом количестве доступных ресурсов целессообразнее может быть использование бэкфиллинга или МПБ (они обеспечиваю примерно одинаковое значение показателя), однако при увеличении масштабов вычислительной среды, более эффективным становится подход с использованием ЦСП.

Аналогичный вывод можно сделать, если оценивать эффективность использования ресурсов на основе среднего времени завершения заданий (рис. 8).

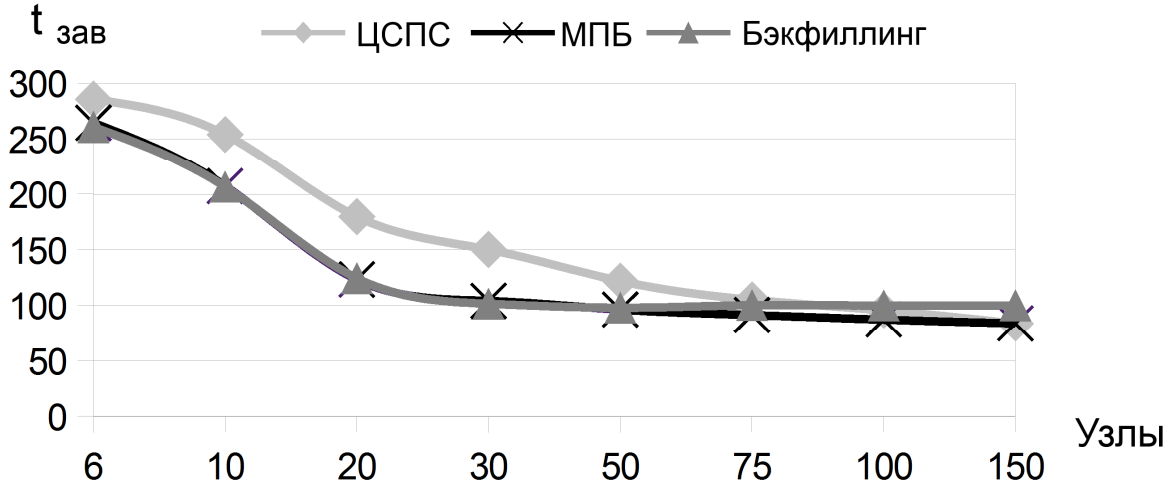


Рис. 8. Среднее время завершения заданий пакета

Заключение

В работе рассмотрены стратегии и эвристики планирования с различными целевыми критериями, основанные на модели справедливого разделения ресурсов, учитывающей предпочтения всех участников ВО. Предложен подход к решению задачи справедливого разделения ресурсов между участниками ВО. Показано преимущество МПСР по срав-

нению с исходной циклической схемой планирования. Владельцы для получения максимальной прибыли или необходимого уровня загрузки среды могут варьировать удельную стоимость предоставления ресурсов (например, предлагать скидки). На основе сочетания циклической схемы и бэкфиллинга предложен комбинированный подход (эвристика МПБ), который демонстрирует компромиссные результаты по сравнению с МПСР и бэкфиллингом. Удается эффективно использовать доступные ресурсы (за счет применения бэкфиллинга) при эффективном выполнении части потока заданий в ВО (за счет оптимизации в МПСР). Показана состоятельность планов, составленных МПСР в условиях, когда фактическое время выполнения заданий может существенно отличаться от пользовательской оценки.

Полученные результаты используются в проекте создания прототипа метапланировщика – программного комплекса планирования потоков заданий и управления неотчуждаемыми ресурсами в распределенных вычислительных средах, основные компоненты которого представлены в [19–21].

Работа выполнена при содействии Совета по грантам Президента Российской Федерации для поддержки ведущих научных школ (шифр НШ-362.2014.9), РФФИ (проект 12-07-00042), Минобрнауки России, задание № 2014/123 на выполнение государственных работ в сфере научной деятельности в рамках базовой части государственного задания (проект 2268).

Литература

1. Garg S.K., Buyya R., Siegel H.J. Scheduling Parallel Applications on Utility Grids: Time and Cost Trade-off Management // 32nd Australasian Computer Science Conference, Wellington, New Zealand, Proceedings. 2009. P. 151-159.
2. Degabriele J.P., Pym D. Economic Aspects of a Utility Computing Service, Bristol, July 3, 2007. 23 p. (Technical Report, Trusted Systems Laboratory, HP Laboratories, HPL-2007-101).
3. Garg S.K., Yeo C.S., Anandasivam A., Buyya R. Environment-conscious Scheduling of HPC Applications on Distributed Cloud-oriented Data Centers // J. Parallel and Distributed Computing. 2011. Vol. 71, No. 6. P. 732-749.
4. Воеводин Вл.В., Жолудев Ю.А., Соболев С.И., Стефанов К.С. Эволюция системы метакомпьютинга X-Com // Вестник Нижегородского университета им. Н.И. Лобачевского. 2009. № 4. С. 157–164.
5. Kurowski K., Nabrzyski J., Oleksiak A., Weglarz J. Multicriteria Aspects of Grid Resource Management. In: Nabrzyski, J., Schopf, J.M., Weglarz J. (eds.). Grid resource management. State of the art and future trends. Kluwer Academic Publishers, Boston, 2003. P. 271–293.
6. Garg S.K., Konugurthi P., Buyya R. A Linear Programming-driven Genetic Algorithm for Meta-scheduling on Utility Grids // J. Par., Emergent and Distr. Systems. 2011. Vol. 26. P. 493–517.
7. Buyya R., Abramson D., Giddy J. Economic Models for Resource Management and Scheduling in Grid Computing // J. Concurrency and Computation. 2002. Vol. 14, No. 5. P. 1507–1542.

8. Ernemann C., Hamscher V., Yahyapour R. Economic Scheduling in Grid Computing // Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.), JSSPP 2002. LNCS, Vol. 2537, Springer, Heidelberg, 2002. P. 128–152.
9. Lee Y.C., Wang C., Zomaya A.Y., Zhou B.B. Profit-driven Scheduling for Cloud Services with Data Access Awareness // J. Par. and Distr. Computing. 2012. Vol. 72, No. 4. P. 591–602.
10. Toporkov V.V. Job and Application-Level Scheduling in Distributed Computing // Ubiquitous Computing and Communication J. Applied Computing. 2009. Vol. 4, No. 3. P. 559 – 570.
11. Toporkov V.V., Toporkova A., Tselishchev A., Yemelyanov D. Job and Application-Level Scheduling: an Integrated Approach for Achieving Quality of Service in Distributed Computing // 4th International Conference on Dependability of Computer Systems, Proceedings. IEEE CS Press, Los Alamitos, 2009. P. 202 – 209.
12. Aida K., Casanova H. Scheduling Mixed-parallel Applications with Advance Reservations // 17th IEEE Int. Symposium on HPDC, Proceedings. IEEE CS Press, New York. 2008. P. 65–74.
13. Ando S., Aida K. Evaluation of Scheduling Algorithms for Advance Reservations // Information Processing Society of Japan SIG Notes. HPC-113, 2007. P. 37-42.
14. Elmroth E., Tordsson J. A Standards-based Grid Resource Brokering Service Supporting Advance Reservations, Coallocation and Cross-Grid Interoperability // J. of Concurrency and Computation. 2009. Vol. 25, No. 18. P. 2298–2335.
15. Toporkov V. Packet Task Processing in Distributed Computational Environments with Inalienable Resources // Automation and Remote Control, 2012. Vol. 73, No. 10. P. 1632–1647.
16. Toporkov V., Tselishchev A., Yemelyanov D., Bobchenkov A. Dependable Strategies for Job-flows Dispatching and Scheduling in Virtual Organizations of Distributed Computing Environments // Complex Systems and Dependability, Proceedings. AISC, vol. 170, Springer, Heidelberg, 2012. P. 240–255.
17. Jackson D., Snell Q., Clement M. Core Algorithms of the Maui Scheduler // LNCS, vol. 2221, Springer, Heidelberg, 2001. P. 87-102.
18. Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D. Slot Selection Algorithms for Economic Scheduling in Distributed Computing with High QoS Rates // New Results in Dependability and Computer Systems. AISC, vol. 224, Springer, Heidelberg, 2013. P. 459-468.
19. Свидетельство о государственной регистрации программы для ЭВМ № 2013613799 «Модуль комбинированного планирования потока заданий «Batch Slice Filling». Правообладатель: НИУ «МЭИ» (RU). Авторы: Топорков В.В., Емельянов Д.М. 16.04.2013 г.

Топорков Виктор Васильевич, д.т.н., заведующий кафедрой вычислительной техники, Национальный исследовательский университет «МЭИ» (Москва, Российская Федерация), ToporkovVV@mpei.ru

Бобченков Александр Викторович., к.т.н., младший научный сотрудник кафедры вычислительной техники, Национальный исследовательский университет «МЭИ» (Москва, Российская Федерация), groddenator@gmail.com

Емельянов Дмитрий Михайлович, к.т.н., ассистент кафедры вычислительной техники, Национальный исследовательский университет «МЭИ» (Москва, Российская Федерация), Yemelyanov.Dmitry@gmail.com

Целищев Алексей Сергеевич, к.т.н., Европейская организация по ядерным исследованиям (CERN) (Женева, Швейцария), Alexey.Tselishchev@cern.ch

Поступила в редакцию 9 апреля 2014 г.

Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 2, pp. 43–62

SCHEDULING METHODS AND HEURISTICS IN DISTRIBUTED COMPUTING WITH NON-DEDICATED RESOURCES

V.V. Toporkov, Moscow Power Engineering Institute (Moscow, Russian Federation),
A.V. Bobchenkov, Moscow Power Engineering Institute (Moscow, Russian Federation),
D.M. Yemelyanov, Moscow Power Engineering Institute (Moscow, Russian Federation),
A.S. Tselishchev, European Organization for Nuclear Research (CERN) (Geneva, Switzerland)

The paper is devoted to the questions of effective computations scheduling in virtual organizations with non-dedicated resources, including utility Grid. A complex combination of job-flow and application-level scheduling methods and heuristics, providing virtual organization resources share and allocation policies based on the economic principles is studied.

Keywords: distributed computing, scheduling, optimization, slot, alternative, cyclic scheduling scheme.

References

1. Garg S.K., Buyya R., Siegel H.J. Scheduling Parallel Applications on Utility Grids: Time and Cost Trade-off Management // 32nd Australasian Computer Science Conference, Wellington, New Zealand, Proceedings. 2009. P. 151-159.
2. Degabriele J.P., Pym D. Economic Aspects of a Utility Computing Service, Bristol, July 3, 2007. 23 p. (Technical Report, Trusted Systems Laboratory, HP Laboratories, HPL-2007-101).
3. Garg S.K., Yeo C.S., Anandasivam A., Buyya R. Environment-conscious Scheduling of HPC Applications on Distributed Cloud-oriented Data Centers // J. Parallel and Distributed Computing. 2011. Vol. 71, No. 6. P. 732-749.
4. Voevodin Vl.V., Zholudev Y.A., Sobolev S.I., Stefanov K.S. Evoluzia sistemy metacomputinga X-Com [X-Com metacomputing system evolution] // Publishing of the N. I. Lobachevsky State University of Nizhny Novgorod 2009. No. 4. P. 157–164.
5. Kurowski K., Nabrzyski J., Oleksiak A., Weglarz J. Multicriteria Aspects of Grid Resource Management. In: Nabrzyski, J., Schopf, J.M., Weglarz J. (eds.). Grid resource management. State of the art and future trends. Kluwer Academic Publishers, Boston, 2003. P. 271–293.

6. Garg S.K., Konugurthi P., Buyya R. A Linear Programming-driven Genetic Algorithm for Meta-scheduling on Utility Grids // J. Par., Emergent and Distr. Systems. 2011. Vol. 26. P. 493–517.
7. Buyya R., Abramson D., Giddy J. Economic Models for Resource Management and Scheduling in Grid Computing // J. Concurrency and Computation. 2002. Vol. 14, No. 5. P. 1507–1542.
8. Ernemann C., Hamscher V., Yahyapour R. Economic Scheduling in Grid Computing // Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.), JSSPP 2002. LNCS, Vol. 2537, Springer, Heidelberg, 2002. P. 128–152.
9. Lee Y.C., Wang C., Zomaya A.Y., Zhou B.B. Profit-driven Scheduling for Cloud Services with Data Access Awareness // J. Par. and Distr. Computing. 2012. Vol. 72, No. 4. P. 591–602.
10. Toporkov V.V. Job and Application-Level Scheduling in Distributed Computing // Ubiquitous Computing and Communication J. Applied Computing. 2009. Vol. 4, No. 3. P. 559 – 570.
11. Toporkov V.V., Toporkova A., Tselishchev A., Yemelyanov D. Job and Application-Level Scheduling: an Integrated Approach for Achieving Quality of Service in Distributed Computing // 4th International Conference on Dependability of Computer Systems, Proceedings. IEEE CS Press, Los Alamitos, 2009. P. 202 – 209.
12. Aida K., Casanova H. Scheduling Mixed-parallel Applications with Advance Reservations // 17th IEEE Int. Symposium on HPDC, Proceedings. IEEE CS Press, New York. 2008. P. 65–74.
13. Ando S., Aida K. Evaluation of Scheduling Algorithms for Advance Reservations // Information Processing Society of Japan SIG Notes. HPC-113, 2007. P. 37-42.
14. Elmroth E., Tordsson J. A Standards-based Grid Resource Brokering Service Supporting Advance Reservations, Coallocation and Cross-Grid Interoperability // J. of Concurrency and Computation. 2009. Vol. 25, No. 18. P. 2298–2335.
15. Toporkov V. Packet Task Processing in Distributed Computational Environments with Inalienable Resources // Automation and Remote Control, 2012. Vol. 73, No. 10. P. 1632–1647.
16. Toporkov V., Tselishchev A., Yemelyanov D., Bobchenkov A. Dependable Strategies for Job-flows Dispatching and Scheduling in Virtual Organizations of Distributed Computing Environments // Complex Systems and Dependability, Proceedings. AISC, vol. 170, Springer, Heidelberg, 2012. P. 240–255.
17. Jackson D., Snell Q., Clement M. Core Algorithms of the Maui Scheduler // LNCS, vol. 2221, Springer, Heidelberg, 2001. P. 87-102.
18. Toporkov V., Toporkova A., Tselishchev A., Yemelyanov D. Slot Selection Algorithms for Economic Scheduling in Distributed Computing with High QoS Rates // New Results in Dependability and Computer Systems. AISC, vol. 224, Springer, Heidelberg, 2013. P. 459-468.

Received 9 April 2014.

УДК 004.021

ОЦЕНКА ПОПУЛЯРНОСТИ АВТОРОВ СОЦИАЛЬНОЙ СЕТИ С ПОМОЩЬЮ ПОИСКА ЭКСПЕРТОВ НА ПРИМЕРЕ СЕРВИСА TWITTER

Р.М. Миниахметов, Е.О. Цацина

В данной работе рассмотрена смешанная языковая модель, применяемая для поиска экспертов в таких областях как анализ социальных сетей и информационный поиск, предложена адаптация этой модели для социальной сети Twitter. Рассмотрены метрики популярности в социальной сети Twitter. Предложена формула оценки популярности пользователей социальной сети Twitter с учетом оценки релевантности их сообщений заданной теме, а также описан прототип системы для сбора данных и оценки популярности по предложенной формуле.

Ключевые слова: анализ социальных сетей, информационный поиск, интеллектуальный анализ данных, поиск экспертов, анализ популярности.

Введение

В настоящее время одной из актуальных задач в области анализа социальных сетей и информационного поиска является задача поиска экспертов. Задача поиска экспертов заключается в нахождении пользователей, сообщения которых более релевантны заданной теме [1]. Большинство подходов, применяемых в задаче поиска экспертов при анализе социальных сетей, не учитывают историю сообщений пользователя и основываются только на анализе профиля пользователя или его связей в социальном графе [1, 2].

На основе поиска экспертов в данной работе предложена формула оценки популярности пользователей социальной сети Twitter. Предложенная формула, в отличие от существующих сервисов для оценки популярности, учитывает не только базовые метрики популярности социальной сети Twitter, но и содержание (контекст) сообщений пользователей.

Рассмотренный в работе подход — использование экспертного балла автора, будет более интересен и полезен для пользователя, который захочет подписаться на сообщения автора интересующих его тем, а не на автора, который просто имеет много подписчиков.

Статья организована следующим образом. В разделе 1 рассмотрены смешанная языковая модель для поиска экспертов и метрики популярности для социальной сети Twitter. В разделе 2 предложены адаптация смешанной языковой модели, применяемой в задаче поиска экспертов, для социальной сети Twitter и формула для оценки популярности в этой сети, описан прототип системы для сбора необходимых данных и вычисления оценки популярности по предложенной формуле. В разделе 3 представлены результаты вычислительных экспериментов. В заключении суммируются основные результаты работы и описываются направления дальнейших исследований.

1. Анализ предметной области

1.1. Смешанная модель для поиска экспертов

В информационном поиске выделяют языковые модели (query likelihood language model) [5], которые описывают зависимость между текстом документа и поисковым за-

просом как вероятность появления в документе d термов t из запроса q:

$$P(d \mid q) = P(q \mid d)P(d) \tag{1}$$

Предполагается, что появление термов t_i в запросе q — независимое событие:

$$P(q \mid d) = \prod_{t \in q} P(t_i \mid d) \tag{2}$$

В данной работе для нахождения экспертов была выбрана смешанная языковая модель [3], которая определена следующим образом.

Пусть между запросом q и документом d_j существует семантический слой в виде набора тем $\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$. Каждая тема θ_m семантически связана как с запросами, так и с документами. Также, каждый документ и запрос связаны с множеством тем. В этом случае связь между запросом и документами моделируется не напрямую, а через связь с темами. Таким образом, получим:

$$P(q \mid d_j) = \sum_{m=1}^{k} P(q \mid \theta_m)P(\theta_m \mid d_j), \tag{3}$$

где $P(q \mid \theta_m)$ — вероятность того, что запрос q относится к теме θ_m, а $P(\theta_m \mid d_j)$ — вероятность того, что тема θ_m относится к документу d_j.

Подсчитать с какой вероятностью запрос q относится к автору e можно при помощи следующей формулы:

$$P(q \mid e) = \sum_{d_j \in D_e} P(q \mid d_j)P(d_j \mid e) = \sum_{d_j = D_e} \sum_{m=1}^{k} \prod_{t_i \in q} P(t_i \mid \theta_m)P(\theta_m \mid d_j)P(d_j \mid e) \tag{4}$$

Смешанная языковая модель для определения экспертов используется дополнительный параметр — тему. Данная модель не связывает термы из запроса и сообщения автора напрямую, тем самым не предполагает появление всех термов из запроса в сообщениях авторов. Это является ключевым фактором выбора данной модели для адаптации и применения на данных социальной сети Twitter, так как максимальный размер сообщения в социальной сети Twitter составляет всего 140 символов и вероятность появления всех термов из запроса в таком коротком сообщении невелика.

Данная модель позволяет выделить темы из множества сообщений авторов и связать запрос с более релевантной темой, что помогает пользователю выбирать наиболее интересных авторов социальной сети.

1.2. Метрики популярности в социальной сети Twitter

Социальная сеть Twitter — это сервис для публичного обмена короткимим сообщения. Отношения между пользователями в Twitter несколько отличаются от других сетей, часто однонаправлены и базируются на совместных интересах. Пользователь может стать подписчиком определенного автора, и публичные сообщения этого автора будут отображаться в новостной ленте пользователя. Благодаря этой функции многие известные и популярные люди имеют аккаунт в этой социальной сети.

Одной из практически полезных задач в анализе социальных сетей является оценка популярности пользователей [7, 8]. Такая оценка помогает рекомендательным сервисам точнее

подбирать содержание сообщений и дает пользователям простой и быстрый способ для поиска нужной информации.

Для оценки популярности в сети Twitter используются различные метрики, основанные на следующих базовых параметрах самой сети Twitter:

1. Количество подписчиков пользователя (followers).
2. Количество повторных размещений сообщений (retweet) пользователя другими пользователями.
3. Количество отметок «понравилось» (favourites, likes), полученных от других пользователей у добавленного автором сообщения.

Проведенный анализ рейтингов популярности в сети Twitter показал, что большинство рейтингов, подобных Twitaholic [15], используют только базовые метрики сети Twitter или их комбинацию, но не учитывают информацию, содержащуюся в сообщениях пользователя.

Другой метрикой (не базовой), которую авторы работы [8] применили для анализа сети Twitter, является определение связей между авторами сети и их подписчиками. Связь рассматривается как наличие перекрестных подписок между авторами сети в рамках одной темы. Такая метрика используется при анализе многих социальных сетей, однако, не учитывает насколько автор связан с темами его сообщений. Степень связи автора и тем его сообщений может быть более интересна для тех пользователей сервиса, которые хотят получать больше информации по определенной теме.

В данной работе вводится такая метрика как экспертный балл, которая показывает, насколько автор является экспертом по интересующей пользователя теме.

2. Оценка популярности авторов в социальной сети Twitter на основе поиска экспертов

В данном разделе приводится описание смешанной модели, применяемой для поиска экспертов, на основе работы [17] и адаптация данной модели для социальной сети Twitter.

2.1. Определение тем сообщений автора

Пусть d_i — сообщение автора e в наборе сообщений D. Каждое сообщение d_i представляет набор термов (ключевых слов) $T_i = \{t_{i1}, t_{i2}, ..., t_{ik}\}$. Будем считать, что термы встречаются в сообщениях со схожим содержанием, если они часто появляются в наборах термов T_i. Определим тему θ для сети Twitter как набор независимых термов, которые часто встречаются в сообщениях одного автора.

Для нахождения часто встречающихся наборов термов использовался алгоритм FP-Growth [18].

Темы сообщений пользователя определяются на основе найденных шаблонов L_i. Для определения темы используется алгоритм кластеризации графов на основе клик:

1. Каждый шаблон $L_i = \{t'_{i1}, ..., t'_{ik}\}$ представляет собой вершину v_i графа G так, что $v_i = L_i$.
2. Если шаблоны имеют общие термы, то будем считать, что между вершинами графа, представляющего данные наборы термов, есть ребро.
3. Найдем все максимальные полные подграфы (клики) в графе G.
4. Для каждого подграфа g_i в графе G, необходимо объединить шаблоны L_i соответствующие каждой вершине подграфа.

Рассмотрим следующий пример. Пусть имеется подграф $g_i = \{v_1, v_3\}$, где $v_1 = L_1 = \{show, night, music\}$, а $v_3 = L_3 = \{new, dance, show\}$. Тогда темой будет объединение соответсвующих вершинам шаблонов L_1 и L_3, такое что $\theta_m = L_1 \cup L_3 = \{show, night, music\} \cup \{new, dance, show\} = \{show, night, music, new, dance\}$.

2.2. Определение авторов, сообщения которых более релевантны заданным темам

Пусть e — автор сообщений в Twitter и $q = \{t_{i1}, t_{i2}, ..., t_{ik}\}$ — запрос по теме, представленный набором термов и $d_i = \{t_{i1}, t_{i2}, ..., t_{ik}\}$ — сообщение автора. К первоначальному набору термов в запросе q будут добавлены термы из тех тем авторов, которые определяются как релевантные запросу [8]. Пусть $expert_score(e, q)$ — экспертный балл автора e по теме q, который отражает насколько релевантны сообщения автора заданной теме. Тогда формула нахождения экспертов определена следующим образом:

$$expert_score(e, q) = P(e \mid q) = P(e) \sum_{\theta_m \in \Theta} \prod_{t_i \in q} P(t_i \mid \theta_m) P(\theta_m \mid e), \tag{5}$$

где $P(e)$ определяет вероятность появления термов из запроса q в сообщениях пользователя e.

Подсчет вероятностей $P(t_i \mid \theta_m)$, $P(\theta_m \mid e)$ производится с помощью модели скрытого распределения Дирихле (Latent Dirichlet Allocation) [19]:

$$P(e) = \frac{\sum_{t_i \in q, d_i \in D_e} f(t_i, d_i)}{\mid D_e \mid}, \; f(t_i, d_i) = \begin{cases} 0, & \text{если } t_i \cap d_i = 0 \\ 1, & \text{в противном случае,} \end{cases} \tag{6}$$

$$P(t_i \mid \theta_m) = \frac{count(t_i^{(\theta_m)}) + \beta}{count(t^{(\theta_m)}) + |T|\beta}, \quad P(\theta_m \mid e) = \frac{count(\theta_m^{(e)}) + \alpha}{count(\theta^{(e)}) + |\Theta|\alpha}, \tag{7}$$

где $count(t_i^{(\theta_m)})$ — частота появления терма $t_i \in \theta_m$ во всех сообщениях авторов, $count(t^{(\theta_m)})$ — частота появления всех термов из темы $\theta_m = \{t_1, ..., t_n\}$, $|T|$ — общее количество термов (слов) во всех сообщениях автора, $count(\theta_m^{(e)})$ — частота появления термов из темы θ_m во всем массиве документов автора D_e, $count(\theta^{(e)})$ — частота появления термов из всего набора тем $\Theta = \{\theta_1, ..., \theta_n\}$ во всем массиве документов автора D_e, $|\Theta|$ — общее количество тем авторов. Параметры α и β — априорные вероятности распределения тем относительно документов и термов относительно тем соответственно.

2.3. Формула популярности авторов

В результате анализа метрик была выявлена сильная корреляция между повторными размещениями сообщений автора (retweets) и метками «понравилось» у сообщений автора (favourites, likes), поэтому они объединены в одну метрику rt_likes.

Для получения формулы оценки популярности был проведен линейный регрессионный анализ базовых метрик сети Twitter и экспертного балла. Будем считать, что чем больше значения базовых метрик и экспертного балла, тем выше соответствующая оценка популярности. Тогда предполагаемое значение популярности для проведения множественной регрессии от трех независимых параметров определяется как модуль радиус-вектора в пространстве метрик $expert_score$, rt_likes_score, $followers_score$.

Таким образом, формула популярности с учетом экспертного балла имеет следующий вид:

$$popularity(e,q) = r_1 \cdot followers_score(e) + r_2 \cdot rt_likes_score(e) + r_3 \cdot expert_score'(e,q), \quad (8)$$

где r_1, r_2, r_3 — коэффициенты множественной регрессии.

Относительное количество подписчиков автора считается по формуле:

$$followers_score(e) = \frac{followers_num(e)}{\sum\limits_{e} followers_num(e)} \quad (9)$$

Относительное количество повторных размещений сообщений и меток «понравилось» у сообщений автора, которые соответствуют теме запроса, определяется следующим образом:

$$rt_likes_score(e) = \frac{\sum_{d_i \in D_e} rt_num(d_i) + likes_num(d_i)}{\sum\limits_{e} \sum_{d_i \in D_e} rt_num(d_i) + likes_num(d_i)}, d_i \cup q \neq 0 \quad (10)$$

Нормированное значение экспертного балла вычисляется по следующей формуле:

$$expert_score'(e,q) = \frac{expert_score(e)}{\sum\limits_{e} expert_score(e)} \quad (11)$$

2.4. Прототип системы расчета оценки популярности

Для вычисления оценки популярности авторов в социальной сети Twitter на основе поиска экспертов был разработан прототип программной системы. На рис. 1 изображен технологический цикл разработанного прототипа. Каждый этап работы алгоритма выполняется отдельной подсистемой, реализованных в виде утилит на языке Python.

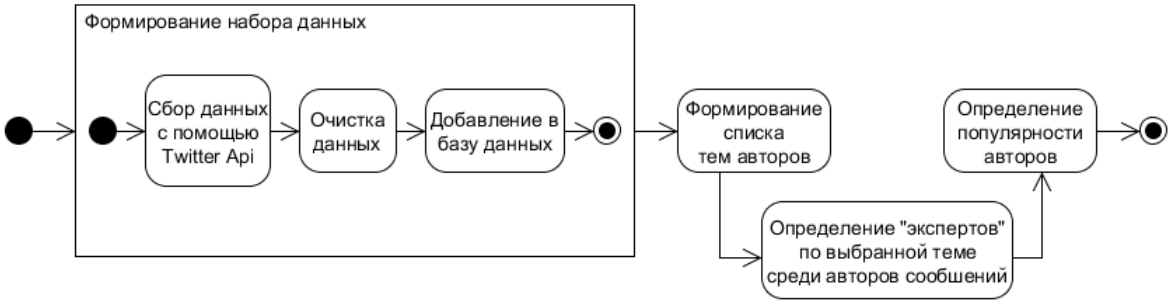


Рис. 1. Технологический цикл работы прототипа системы

Сбор данных реализован с помощью функций библиотеки tweepy [9] для языка Python, которая позволяет взаимодействовать с Twitter REST API [10]. С помощью REST API социальная сеть Twitter предоставляет доступ ко всем сущностям сети. Для сбора данных были использовныы следующие функции REST API:

1. $get_user(id|screen_name)$ возвращает всю доступную информацию (например, географическое положение, число подписчиков) о пользователе по указанному идентификатору или уникальному имени пользователя. Для получения информации об авторах был составлен список имен наиболее популярных авторов сети Twitter согласно рейтингу популярности Twitaholic [15].
2. $user_timeline(id)$ возвращает список сообщений автора по его идентификатору.

Очистка полученных данных представляет собой следующие действия:

1. Отбрасывание сообщений, которые содержат символы не из таблицы символов ASCII, так как анализируются только англоязычные сообщения.
2. Отбрасывание авторов, которые имееют большой процент повторяющихся (одинаковых) сообщений.

Расчет коэффициентов в формуле популярности 8 производится на обучающей выборке, которая представляет собой информации об авторах, их сообщениях и темах. Для хранения обучающей выборки была использована свободная СУБД PostgreSQL [11]. Схема базы данных изображена на рис. 2.

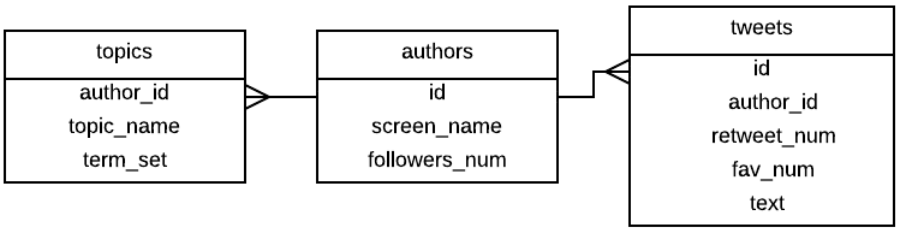


Рис. 2. Схема базы данных

Для формирования списка тем авторов используется алгоритм, описанный в подразделе 2.1. Определенные темы добавляются в базу данных и используются для расчета экспертного балла.

Определение авторов, сообщения которых релевантны заданной теме, выполняется по алгоритму из подраздела 2.2. Экспертный балл для каждого автора вычисляется по формуле 5. Результатом работы данной подсистемы является файл формата CSV, который содержит идентификатор автора и соответствующее ему значение экспертного балла.

Расчет оценки популярности авторов ведется по формуле 8. Для построения линейной регрессионной модели от трех переменных использовалась функция $LinearRegression()$, реализующая метод наименьших квадратов для построения регресии, из библиотеки scikit-learn [16] для языка Python. Результатом работы данной подсистемы является файл формата CSV, который содержит идентификатор автора и соответствующую ему оценку популярности.

Главным недостатком реализованного прототипа является достаточно продолжительное время работы на обучающей выборке, которая представляет собой большой набор данных. Возможны две основные причины этого недостатка:

1. Сбор и хранение больших объемов данных — запрос и передача данных по сети через REST API.
2. Расчет по смешанной языковой модели и модели линейной регрессии на больших объемах данных.

Среди возможных эффективных путей устранения указанного недостатка прототипа могут быть следующие:

1. Реализация онлайн-алгоритма [12] поиска экспертов и расчета оценки популярности, который не будет требовать хранения больших объемов данных.
2. Использование СУБД, которые способны эффективно обрабатывать и хранить большие объемы данных [14].
3. Оптимизация сбора обучающей выборки на основе репрезентативного сэмплинга [13].

3. Эксперименты

В этом разделе приведена информация о собранных данных для разработанной системы. Приведен результат работы примененной модели поиска экспертов и расчет популярности авторов социальной сети Twitter на основе предложенной формулы популярности. Проведен анализ полученных результатов.

3.1. Собранные данные

Для проверки работы системы было собрано около 12000 сообщений из социальной сети Twitter. Список авторов был составлен на основе рейтинга популярности в социальной сети Twitter [15], который, как и многие подобные рейтинги, строится по количеству подписчиков у автора. Для анализа работы системы были выбраны первые 100 авторов, которые имеют максимальное число подписчиков среди всех пользователей сети Twitter. Информация о собранных данных приведена в табл. 1.

Таблица 1

Информация о собранных данных

	Обучающая выборка	Тестовая выборка
Количество авторов	100	10
Количество сообщений автора	100	200
Всего сообщений	10000	2000

Обучающая выборка использовалась для проведения регрессионного анализа между значением популярности (*popularity*) и относительными значениями метрик сети Twitter ($followers_score$, rt_likes_score), а также экспертным баллом ($expert_score$).

Для проверки адекватности построенной модели были посчитаны средняя ошибка аппроксимации и коэффициент детерминации R^2. Средняя ошибка аппроксимации представляет среднее отклонение расчетных значений от фактических и составляет $0,32\%$, что меньше допустимых 15%. Данное значение свидетельствует об относительной точности построенной модели и о том, что ее можно применять для расчетов. Коэффициент детерминации R^2, характеризующий качество построенной модели, близок к 1 — составляет $0,9999855$ и означает, что модель достаточно точно объясняет поведение прогнозируемого параметра (популярности).

Построенная линейная регрессионная модель позволила оценить зависимость между перечисленными величинами и определить коэффициенты, которые используются при подсчете оценки популярности. На рис. 3–5 изображены графики полученных зависимостей.

График, изображенный на рис. 3 показывает, что зависимость популярности автора от количества подписчиков автора имеет линейный характер. График зависимости популярности автора от значения метрик понравившихся и пересланных сообщений сети Twitter изображен на рис. 4 и представляет собой логарифмическую функцию.

График зависимости между экспертным баллом и популярностью хоть и напоминает отдаленно логарифмическую функцию (рис. 5), имеет большое количество значений, которые плохо описываются этой функцией. Возможными причинами плохого объяснения зависимости популярности от экспертного балла могут являться как неподходящая модель линейной

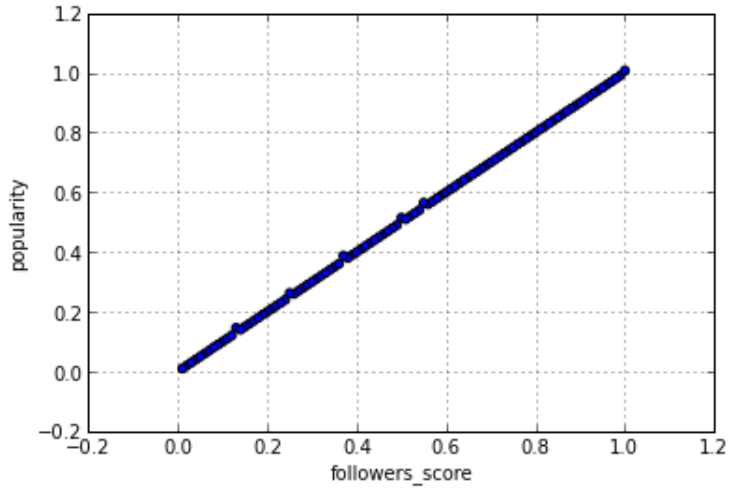


Рис. 3. Зависимость популярности автора от количества подписчиков автора

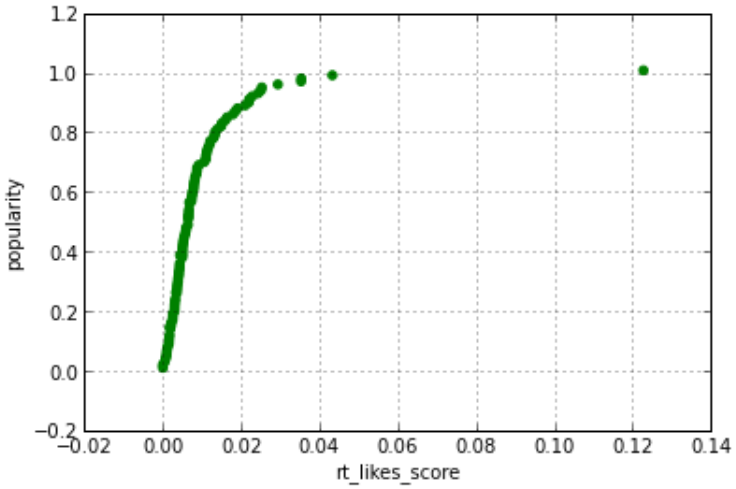


Рис. 4. Зависимость популярности автора от количества повторных размещений сообщений и меток «понравилось» у сообщений автора, соответствующих теме запроса

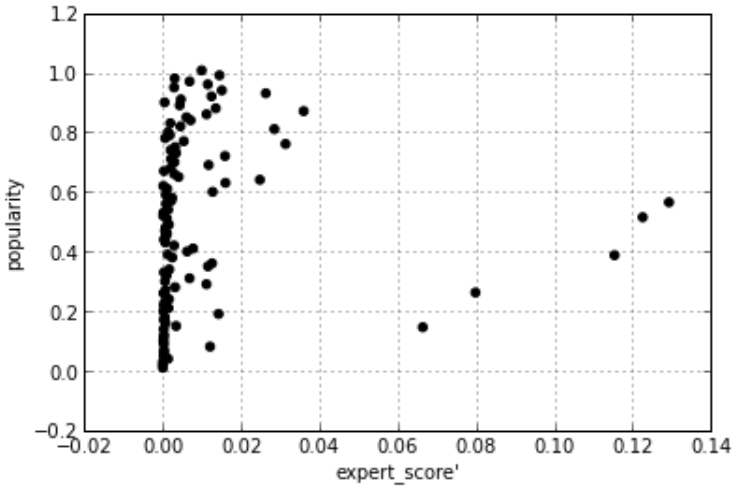


Рис. 5. Зависимость популярности автора от экспертного балла

регрессии, так и недостатки вычисления самой метрики экспертного балла. В то же вре-

мя линейная регрессия хорошо подходит для объяснения зависимости между количеством подписчиков автора и его популярностью. Кроме того, на представленных графиках видно, что значения экспертного балла количественно существенно отличаются от значений числа подписчиков. Таким образом, построенная функция линейной регресии, предсказывающая значение популярности автора по этим трем параметрам, имеет сильную зависимость от количества подписчиков.

3.2. Результаты экспериментов

Далее приведем результаты работы системы. В качестве запроса был выбран один из найденных наборов термов, представляющий определенную тему $q^* =$ *'tonight play last show'*. Алгоритм нахождения тем авторов описан в подразделе 2.1. Результат применения предложенной адаптации смешанной модели поиска экспертов для авторов социальной сети Twitter приведен в табл. 2.

Таблица 2

Позиции авторов относительно релевантности их сообщений запросу q^*

№ п/п	Имя автора	Экспертный балл	Позиция в рейтинге Twitaholic
1	@taylorswift13	0,009396866984837462	5
2	@katyperry	0,007293893139221034	1
3	@TheEllenShow	0,003811523205488189	9
4	@JLo	0,0029136778336937533	10
5	@jtimberlake	0,0027675981458890678	8
6	@ladygaga	0,002531642429819456	4
7	@rihanna	0,0016395838934267649	7
8	@justinbieber	0,0015584543674876422	2
9	@britneyspears	0,0015263600436778333	6
10	@BarackObama	0,00013284391918131865	3

Результаты, приведенные в табл. 2, показывают, что авторы, количество сообщений которых в большей степени соответствовали заданному запросу по теме, занимают первые позиции, несмотря на то, что в выбранном рейтинге популярности занимают последние позиции. Например, автор @JLo переместился с 10 позиции на 4, а @TheEllenShow с 9 на 3, и наоборот, автор @BarakObama опустился с 3 позиции на 10, так как не является представителем массовой культуры и его сообщения в наименьшей степени соответствуют запросу.

В табл. 3 приведены результаты оценки популярности авторов с помощью предложенной в подразделе 8 формулы.

Из табл. 3 видно, что формула оценки популярности зависит как от экспертного балла, так и от метрик популярности, в частности, сильное влияние на распределение авторов оказывает количество подписчиков. Авторы с более высокой позицией в рейтинге популярности Twitaholic оказываются в верхней части полученного списка популярности, однако, автор @taylorswift13, сообщения которого являются более релевантными теме запроса q^* согласно экспертному баллу, также сохранил свою позицию вверху списка популярности.

Таблица 3

Оценка популярности авторов, с учетом экспертного балла по теме q^* и метрик популярности сети Twitter

№ п/п	Имя автора	popularity	Место в рейтинге Twitaholic
1	@katyperry	0,17349268242859633	1
2	@taylorswift13	0,16173712417285024	5
3	@justinbieber	0,16142120932868534	2
4	@ladygaga	0,12378441606619985	4
5	@BarackObama	0,11051037361616957	3
6	@britneyspears	0,10208204786496214	6
7	@rihanna	0,09785773438412347	7
8	@jtimberlake	0,09483032431326485	8
9	@TheEllenShow	0,09016066084936912	9
10	@JLo	0,08371571502915753	10

Таким образом, экспертный балл, даже с учетом большой зависимости популярности от количества подписчиков, оказывает влияние на оценку популярности.

Заключение

В работе представлена смешанная языковая модель для поиска экспертов, адаптированная для социальной сети Twitter, а также предложена формула оценки популярности авторов в социальной сети Twitter с учетом релевантности их сообщений заданной теме. Данная формула оценки популярности авторов учитывает как базовые метрики самой сети Twitter, так и контекст сообщений авторов. Реализован прототип системы, которая сначала ранжирует авторов согласно релевантности их сообщений заданной теме, и на основе полученной экспертной оценки (экспертный балл), производит оценку популярности.

Проведенные вычислительные эксперименты показывают, что предложенная формула оценки популярности имеет сильную зависимость от количества подписчиков автора, однако экспертный балл также влияет на значение популярности автора. Таким образом, авторы, чьи сообщения в большей степени соответствуют теме запроса, могут находится вверху рейтинга популярности, несмотря на то, что они, возможно, менее популярны в соответствии с рейтингом. Предложенная формула оценки популярности помогает найти авторов, которые пишут на определенные темы и также являются достаточно популярными пользователями сети. Формула оценки популярности может быть использована в рекомендательных сервисах для сети Twitter.

На основе реализованной модели поиска экспертов исследования могут быть продолжены по следующим основным направлениям:

1. Поиск экспертов в режиме реальном времени (онлайн-обработка). Например, для рекомендательного сервиса.
2. Полученный экспертный балл может быть использован в качестве базовой оценки в алгоритме поиска экспертов, учитывающем связи авторов [1].

Также исследования могут быть продолжены в направлении улучшения предложенной регрессионной модели для оценки значения популярности: поиск новой модели, которая

будет точнее описывать зависимость экспертного балла автора и его популярности, что позволит увеличить корреляцию значения популярности и сообщений автора по заданной теме и уравновесить по значимости используемые метрики.

Полученные результаты могут быть использованы при поиске ответов на следующие вопросы:

1. Что представляет собой популярность в социальных сетях и как ее оценивать?
2. Какие метрики социальных сетей позволяют точнее и корректнее оценить популярность пользователей?
3. Какие агрегированные значения базовых метрик могут быть использованы при определении популярности?
4. Является ли полученная оценка популярности оценкой реальных пользователей сети или результатом различных способов искусственного увеличения показателей базовых метрик популярности сети?

Литература

1. Zhang, J. Expert Finding in A Social Network / J. Zhang, J. Tang, J. Li // Lecture Notes in Computer Science. — 2007. — Vol. 4443. — P. 1066–1069.

2. Bozzon, A. Choosing the Right Crowd: Expert Finding in Social Networks / A. Bozzon, M. Brambilla, S. Ceri, M. Silvestri, G. Vesci // Proceedings of the 16th International Conference on Extending Database Technology (EDBT '13). — 2013. — P. 637–648.

3. Zhang, J. A Mixture Model for Expert Finding / J. Zhang, J. Tang, L. Liu, J. Li // Proceedings of the 12th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD '08). — 2008. — P. 466–478.

4. Manning, C. Introduction to Information Retrieval / C. Manning, P. Raghavan, H. Schütze — Cambridge University Press, 2008. — 496 p.

5. Ponte, J. A Language Modeling Approach to Information Retrieval / J. Ponte, W. Croft // Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98). —1998. — P. 275–281.

6. Hiemstra, D. A Linguistically Motivated Probabilistic Model of Information Retrieval / D. Heimstra // Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries.— 1998. — P. 569–584.

7. Steck, H. Item Popularity and Recommendation Accuracy / H. Steck // Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries (RecSys '11). — 2011. — P. 125–132.

8. Cha, Y. Incorporating Popularity in Topic Models for Social Network Analysis / Y. Cha, B. Bin, C. Hsieh, J. Cho // Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '13). — 2013. — P. 223–232.

9. Документация Python библиотеки для работы с Twitter API
URL: `http://pythonhosted.org//tweepy/` (дата обращения: 24.04.2014).

10. Документация Twitter REST API
URL: `https://dev.twitter.com/docs/api/1.1` (дата обращения: 24.04.2014).

11. Stonebraker, M. The POSTGRES Next-generation Database Management System / M. Stonebraker, G. Kemnitz // Communications of the ACM. — 1991. — Vol. 34, No. 10. — P. 78–92.

12. Borodin, A. Online Computation and Competitive Analysis / A. Borodin, R. El-Yaniv — Cambridge University Press, 1998. — 432 p.

13. Янцен, Д.Д. Алгоритм репрезентативного сэмплинга для параллельных систем баз данных / Д.Д. Янцен, М.Л. Цымблер // Параллельные вычислительные технологии (ПаВТ'2014): труды международной научной конференции (1–3 апреля 2014 г., г. Ростов-на-Дону). — Челябинск: Издательский центр ЮУрГУ, 2014. — С. 381.

14. Пан, К.С. Подход к разбиению сверхбольших графов с помощью параллельных СУБД / К.С. Пан // Вестник ЮУрГУ. Серия "Вычислительная математика и информатика". — 2012. — № 47(306). Вып. 2. — С. 127–132.

15. Рейтинг популярности пользователей в социальной сети Twitter «Twitaholic» URL: `http://twitaholic.com/` (дата обращения: 24.04.2014).

16. Документация Python библиотеки для интеллектуального анализа данных «scikit-learn» URL: `http://scikit-learn.org/stable/documentation.html` (дата обращения: 24.04.2014).

17. Zhu, H. Finding Experts in Tag Based Knowledge Sharing Communities / H. Zhu, E. Chen, H. Cao // Proceedings of the 5th International Conference on Knowledge Science, Engineering and Management (KSEM '11). — 2011. — P. 183–195.

18. Han, J. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach / J. Han, J. Pei, Y. Yin, R. Mao // Data Mining and Knowledge Discovery. — 2005. — Vol. 8, No. 1. — P. 53–87.

19. Blei, D. Latent Dirichlet Allocation / D. Blei, A. Ng, Y. Yin, M. Jordan // The Journal of Machine Learning Research. — 2003. — Vol. 3. — P. 993–1022.

Миниахметов Руслан Марсович, аспирант, кафедра системного программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), miniakhmetovrm@susu.ac.ru.

Цацина Елизавета Олеговна, студент 4 курса, кафедра системного программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), decemberliz92@gmail.com.

Поступила в редакцию 9 мая 2014 г.

Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 2, pp. 63–76

TWITTER USERS POPULARITY ESTIMATION USING EXPERT FINDING

R.M. Miniakhmetov, South Ural State University (Chelyabinsk, Russian Federation)**,**
E.O. Tsatsina, South Ural State University (Chelyabinsk, Russian Federation)

In this paper we have considered mixed language model that is used for experts finding in areas such as social network analysis and information retrieval, and proposed an adaptation of this model for the social network Twitter. We also have reviewed Twitter popularity metrics and proposed Twitter users' popularity estimation approach based on expert finding, which allows to rank users according to the probability of user being an expert in given query, and have implemented a prototype for data collection and popularity estimation, based on our approach.

Keywords: social network analysis, information retrieval, data mining, expert finding, popularity analysis.

References

1. Zhang J., Tang J., Li J. Expert Finding in A Social Network. Lecture Notes in Computer Science. 2007. Vol. 4443, No. 3. P. 1066–1069.
2. Bozzon A., Brambilla M., Ceri S., Silvestri M., Vesci G. Choosing the Right Crowd: Expert Finding in Social Networks. Proceedings of the 16th International Conference on Extending Database Technology (EDBT '13). 2013. P. 637–648.
3. Zhang J., Tang J., Liu L., Li J. A Mixture Model for Expert Finding. Proceedings of the 12th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD '08). 2008. P. 466–478.
4. Manning C., Raghavan P., Schütze P. Introduction to Information Retrieval. Cambridge University Press, 2008. 496 p.
5. Ponte, J., Croft J. A Language Modeling Approach to Information Retrieval. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98). 1998. P. 275-281.
6. Hiemstra, D. A Linguistically Motivated Probabilistic Model of Information Retrieval. Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries. 1998. P. 569–584.
7. Steck H. Item Popularity and Recommendation Accuracy. Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11). 2011. P. 125–132.
8. Cha Y., Bin B., Cho J. Incorporating Popularity in Topic Models for Social Network Analysis. Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '13). 2013. P. 223–232.
9. Twitter API library for Python documentation
URL: `http://pythonhosted.org//tweepy/` (accessed: 24.04.2014).

10. Twitter REST API Documentation
URL: `https://dev.twitter.com/docs/api/1.1` (accessed: 24.04.2014).

11. Stonebraker M., Kemnitz G. The POSTGRES Next-generation Database Management System // Communications of the ACM. 1991. Vol. 34, No. 10. P. 78–92.

12. Borodin A., El-Yaniv R. Online Computation and Competitive Analysis. Cambridge University Press, 1998. 432 p.

13. Yantsen D.D., Zymbler M.L. Algoritm reprezentativnogo sempling dlya parallel'nykh sistem baz dannykh [A Representative Sampling Method for Parallel DBMS] // Parallel'nyye vychislitel'nyye tekhnologii (PaVT'2014): trudy mezhdunarodnoy nauchnoy konferentsii (1-3 aprelya 2014 g., g. Rostov-na-Donu) [Parallel Computational Technologies (PCT'2014): Proceedings of the International Scientific Conference (April 1-3, 2014, Rostov-on-Don)]. Chelyabinsk, Publishing of the South Ural State University, 2014. P. 381.

14. Pan C.S. Podkhod k razbiyeniyu sverkhbol'shikh grafov s pomoshch'yu parallel'nykh SUBD [An Approach for Very Large Graph Partitioning by Means of Parallel DBMS] // Vestnik YuUrGu. Seriya "Vychislitel'naya matematika i informatika" [Bulletin of the SUSU. Series «Computational Mathematics and Software Engineering»]. 2012. No 47(306). Iss. 2. P. 127–132.

15. Tracking the most popular users of a certain microblogging/social network tool «Twitaholic».
URL: `http://twitaholic.com/` (accessed: 24.04.2014).

16. Machine learning library for Python documentation
URL: `http://scikit-learn.org/stable/documentation.html` (accessed: 24.04.2014).

17. Zhu H., Chen E., Cao H. Finding Experts in Tag Based Knowledge Sharing Communities. Proceedings of the 5th International Conference on Knowledge Science, Engineering and Management (KSEM '11). 2011. P. 183–195.

18. Han J., Pei J., Yin Y., Mao R. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery. 2005. Vol. 8, No. 1. P. 53–87.

19. Blei D., Ng A., Yin Y., Jordan M. Latent Dirichlet Allocation. The Journal of Machine Learning Research. 2003. Vol. 3. P. 993–1022.

Received 9 May 2014.

УДК 621.396.6(07), 537.8(07)

ПРИМЕНЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СХЕМ ПОВЫШЕННОЙ ТОЧНОСТИ В ПРОЕКТИРОВАНИИ АНТЕННЫХ СИСТЕМ

А.Б. **Хашимов,** ***Р.Р. Салихов, Р.С. Альметов***

Предложены эффективные вычислительные методы для исследования математических моделей антенных систем. Для построения математических моделей используются строгие электродинамические принципы и уравнения. Применение квадратурных формул повышенной точности для решения интегрального уравнения Поклингтона и специальной регуляризирующей процедуры для интегрального уравнения II рода позволяет получить устойчивые результаты моделирования с заданной точностью и высокими динамическими характеристиками.

Ключевые слова: математическая модель, интегральные уравнения, квадратурные формулы Гаусса – Лежандра.

Построение эффективных математических моделей (ММ) сложных антенных систем предполагает использование строгих электродинамических принципов и уравнений, а также обоснование используемой формализации граничных условий, оценку погрешностей, вызванных численными методами исследования и конечными ресурсами вычислительной техники. Необходимость строгого электродинамического подхода в задачах проектирования антенных систем вызвана, в первую очередь, необходимостью повышения точности моделирования, так как возрастающие требования к функциональным характеристикам сложных антенных систем (гибридные зеркальные антенны, многоэлементные фазированные антенные решетки, цифровые антенные решетки) приводят к необходимости учета большого числа различных факторов, например, обеспечение параметров электромагнитной совместимости, работа в сложной помеховой обстановке. При этом в ряде случаев помимо требований высокой точности моделирования антенных систем важное практическое значение приобретают вопросы повышенной скорости получения результатов вычислений.

Широко используемый метод интегральных уравнений (ИУ) для построения ММ задач проектирования антенных систем имеет свои особенности и области применения [1]. Метод ИУ является альтернативным методом решения краевых задач, поэтому в одних задачах его применение эффективно, а в других нет. Одним из критериев целесообразности применения ИУ является определенное соответствие электрических размеров исследуемых объектов длине волны возбуждающего электромагнитного поля. Важно отметить, что ИУ формулируются на основе строгих электродинамических уравнений и принципов, например, с использованием леммы Лоренца в интегральной форме. В работе [2] отмечено, что ИУ электродинамики позволяют представить краевую задачу как задачу с нелокальными граничными условиями. Тогда для неоднородных объектов внутреннее поле может быть выражено непосредственно через возбуждающее поле, а поле рассеяния определяется на втором этапе построения решения.

Организация вычислительных схем исследования ММ на основе ИУ требует рационального подхода к формированию систем линейных алгебраических уравнений

(СЛАУ), к которым приводят прямые методы дискретизации ИУ, например, метод моментов [3]. Использование дискретных форм представления решения в виде системы базисных и пробных функций приводит к необходимости выполнения операций численного интегрирования с возможно большей точностью при минимальных временных затратах. В первую очередь это относится к ММ крупноразмерных объектов, когда генерируемая СЛАУ имеет размерность порядка $10^5 \times 10^5$ элементов. В таких задачах быстро нарастающие погрешности округления при выполнении большого количества операций по обработке коэффициентов СЛАУ приводят к появлению неконтролируемых осциллирующих ошибок решения. Фильтрация таких ошибок может быть проведена аппроксимацией решения базисными функциями, отвечающих требованиям физической реализуемости и априорной информации о свойствах решения, например, принадлежности к определенным функциональным пространствам (Соболева, Гельдера) [4].

Так как основной характерной чертой исследования ММ на основе ИУ является редукция к СЛАУ, часто очень большой размерности, то вопросы вычисления, адресации и последовательности обработки коэффициентов СЛАУ приобретают важное практическое значение. При этом неоходимо отметить, что ММ задач проектирования антенных систем имеют векторный комплексный характер. Это приводит к выраженной блочной структуре СЛАУ, так как дискретная форма ИУ предполагает последовательный перебор всех фиксированных точек наблюдения в области построения решения. Для решения СЛАУ могут быть использованы как прямые, так и итерационные методы, выбор которых определяется размерностью задачи и компьютерными ресурсами. Например, использование суперкомпьютерного моделирования для прямых методов позволяет анализировать поле излучения антенных систем, установленных на среднеразмерных аэродинамических объектах. Еще большие возможности появляются при использовании итерационных методов, но здесь быстро нарастает время решения задачи.

Вычисление коэффициентов СЛАУ основано на использовании квадратурных формул – Симпсона, Ньютона – Котеса, Гаусса – Лежандра. Общая оценка погрешности ММ рассматриваемых задач напрямую связана с погрешностями используемых квадратурных формул, следовательно, вопросы выбора и рациональной программной реализации численного интегрирования также имеют важное практическое значение.

Таким образом, разработка вычислительных схем повышенной точности расширяет возможности моделирования сложных антенных систем, когда появляется возможность учета большого количества факторов, влияющих на точность результатов исследования. Такие вычислительные схемы предполагают оптимизацию как методов формирования СЛАУ, так и программной реализации методов численного интегрирования с высокими динамическими свойствами и с заданной точностью.

Рассмотрим некоторые примеры применения вычислительных схем повышенной точности.

1. ММ вибраторных излучателей в тонкопроволочном приближении

Вибраторные излучатели являются одними из наиболее часто используемых при проектировании антенных систем. Простота конструкции, компактность, высокая

технологичность изготовления определяют широкое применение таких излучателей как в виде отдельных антенн, так и в составе сложных многоэлементных систем – ФАР, многолучевых и активных антенных решеток. Детальное изучение ММ вибраторных излучателей позволяет исследовать с высокой степенью достоверности зависимости входного сопротивления в полосе частот с учетом взаимного влияния для сканирующих антенных решеток. Это позволяет проектировать оптимальные согласующие устройства, исследовать поведение ДН, коэффициента направленного действия (КНД) антенной системы в секторе сканирования и заданной полосе частот.

Рассмотрим ММ вибраторного излучателя в тонкопроволочном приближении, когда радиус проводников плеч вибраторов $a \ll \lambda$, где λ – длина волны возбуждающего поля. Строгая электродинамическая формулировка ММ в этом случае приводит к ИУ Поклингтона [5]. В основной декартовой системе координат x, y, z и связанной с ней сферической системой координат R, θ, φ выделим точку (x_0, y_0, z_0), совпадающую с центром симметричного вибраторного излучателя. Введем единичный орт $\vec{i}_d$, определяющий произвольную пространственную ориентацию оси излучателя:

$$\vec{i}_d = \vec{i}_x \cdot \sin\theta_d \cdot \cos\varphi_d + \vec{i}_y \cdot \sin\theta_d \cdot \sin\varphi_d + \vec{i}_z \cdot \cos\theta_d, \qquad (1)$$

где θ_d, φ_d – угловые координаты оси. В точку (x_0, y_0, z_0) поместим начало вспомогательной цилиндрической системы координат ζ, ρ, ψ, где ζ – координата высоты; ρ – координата радиуса; ψ – азимутальная координата точки наблюдения. Выбор этой системы координат обусловлен только внутренней структурой численного решения ИУ Поклингтона, все остальные геометрические операции производятся в основной системе координат. Будем использовать ИУ Поклингтона следующего вида:

$$\int_{-h}^{h} I_d(\zeta') \left[\frac{d^2}{d\zeta^2} + k^2 \right] G(\zeta - \zeta') d\zeta' = \frac{4\pi i k}{Z_c} \vec{i}_d \cdot \vec{E}_i(\zeta), \; -h \le \zeta \le h, \qquad (2)$$

где I_d – распределение линейного тока по оси вибратора, $I_d(\pm h) = 0$; $h = l/2$; l – длина вибратора; $k = 2\pi/\lambda$; $G(\zeta - \zeta') = \exp(-ikr)/r$ – функция, определяющая ядро ИУ; $r = \sqrt{(\zeta - \zeta')^2 + a^2}$, штрих означает принадлежность к точке интегрирования; $\vec{E}_i$ – вектор напряженности электрического поля возбуждения; $Z_c = \sqrt{\mu_a / \varepsilon_a}$ – волновое сопротивление среды. Следуя одной из наиболее часто используемых схем численного решения ИУ (2), рассмотрим следующее представление функции G в выбранной системе координат [5]:

$$G(\zeta, \rho) = \frac{1}{2\pi} \int_0^{2\pi} \frac{e^{-ikr}}{r} d\psi', \qquad (3)$$

где $r = \sqrt{\zeta^2 + \rho^2 + a^2 - 2a\rho\cos\psi'}$. При решении ИУ (2) для тонкопроволочного приближения $\rho = a$. Необходимо отметить, что численное решение ИУ (2) с сильной особенностью требует нахождения значений функции (3) с высокой точностью, особенно для случая $|\zeta| \to 0$, который в основном и определяет устойчивость и точность

всего решения [3]. Производя замену переменных в (3) для случая $\rho = a$ ядро ИУ можно представить следующим образом [5]:

$$G(\zeta) = \frac{2\mathrm{K}(\kappa)}{\pi r_{\max}} \int_0^1 \frac{\exp\left(-ikr_{\max}\sqrt{1-\kappa^2\sin^2\theta}\right)}{\sqrt{1-\kappa^2\cdot\sin^2\theta}} d\theta, \tag{4}$$

где $\mathrm{K}(\kappa)$ – полный эллиптический интеграл I рода; $\kappa = 2a\Big/\sqrt{\zeta^2+4a^2}$; $r_{\max} = 2a/\kappa$; $\theta = (\psi' - \pi)/2$. Такое представление дает возможность получить асимптотическую оценку для случая $|\zeta| \to 0$:

$$G(\kappa) \approx \frac{1}{\pi a} \ln \frac{8a}{|\zeta|}. \tag{5}$$

Далее производится аппроксимация второй производной в ядре ИУ методом конечных разностей:

$$\frac{d^2}{d\zeta^2} G(\zeta_m) = \frac{G(\zeta_{m-1}) - 2G(\zeta_m) + G(\zeta_{m+1})}{\Delta l^2} + o\left(\Delta l^2\right), \tag{6}$$

где Δl – интервал разбиения длины плеча вибраторного излучателя. Необходимо отметить, что критерии корректности численного решения ИУ предполагают одинаковый порядок погрешности вычислительных процедур, поэтому разные оценки погрешности при вычислении (5), (6) для произвольного расположения точек наблюдения и источников затрудняют получение общей оценки погрешности численного решения ИУ (2). Кроме того, формирование СЛАУ потребует в этом случае дополнительного блока для определения случая совпадения этих точек, что приводит к увеличению времени вычислений.

Рассмотрим вариант прямого вычисления коэффициентов СЛАУ с использованием квадратурных формул Гаусса – Лежандра повышенной точности без выделения особенностей (5) и без аппроксимации (6). В этом случае прямые вычисления производятся по одной и той же однородной схеме с общим порядком погрешности. Вычислительные эксперименты показывают, что квадратурные формулы Гаусса – Лежандра с числом узлов 32 (четное количество узлов гарантирует отсутствие особенности при $\zeta = \zeta'$) обеспечивают высокую устойчивость численного решения при малых затратах компьютерного времени. Программная реализация квадратурной формулы Гаусса – Лежандра построена для инвариантного интервала $[-1,1]$ с использованием программы *legendre* системы MATLAB. На рис. 1 приведены результаты расчета распределения тока симметричного вибратора длиной $l = 0.485\lambda$, радиус $a = 0,007\lambda$. Для объективной оценки эффективности вычислительной схемы повышенной точности дискретизация ИУ (2) проведена для одного и того же возбуждающего поля в зазоре вибратора в виде δ-функции, и для одного и того же набора базисных функций метода моментов.

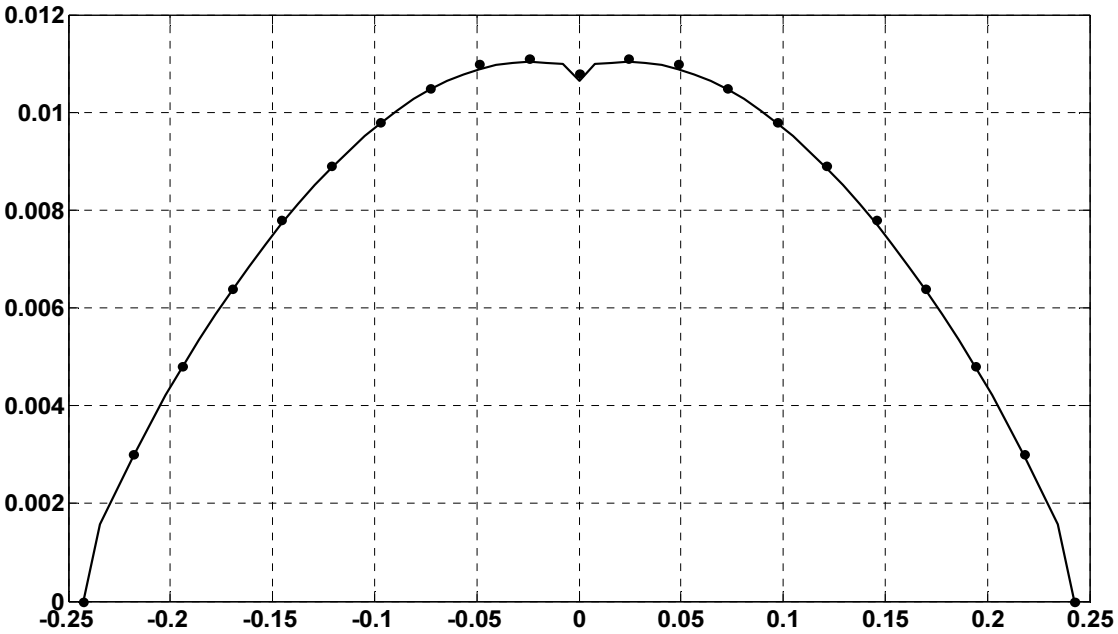


Рис. 1. Распределение тока вибраторного излучателя

Вычислительные эксперименты показывают, что при одних и тех же исходных данных вычислительная схема повышенной точности и рациональной организации формирования СЛАУ обеспечивает уменьшение времени расчетов до 20%. Следовательно, при решении более сложных задач, например синтез ДН многоэлементной ФАР с вибраторными излучателями, когда в целевой функции на каждом шаге минимизации необходимо рассчитывать токовые распределения вибраторов с учетом взаимного влияния, такая экономия компьютерного времени может существенно уменьшить общее время расчетов.

2. Двумерные задачи рассеяния электромагнитных полей на идеально проводящих объектах

В задачах проектирования антенных систем часто возникает необходимость учета влияния объекта установки (башни и мачты радио-телевизионных передающих центров, телекоммуникационных систем) на ДН в вертикальной и горизонтальной плоскости. В работе [6] показано, что для антенных систем вертикальной поляризации, размещаемых вблизи системы параллельных однородных протяженных цилиндрических объектов, оси которых параллельны плоскости поляризации антенной системы, асимптотическая оценка ДН в горизонтальной плоскости для дальней зоны может быть определена через поле рассеяния плоской волны на этой системе цилиндрических объектов. Практическая значимость такого подхода объясняется тем, что решение трехмерной задачи рассеяния возбуждающего поля сложной антенной системы на протяженных цилиндрических объектах намного сложнее решения двумерной задачи рассеяния плоской волны на контурах, образованных сечением цилиндрических объектов горизонтальной

плоскостью. Кроме того, при проектировании антенной системы заданные требования к ДН предъявляются именно для дальней зоны. С помощью такого подхода можно реализовать решение достаточно сложных инженерных задач [7]. Таким образом, решение двумерной задачи рассеяния плоской волны имеет важной практическое значение.

Рассмотрим задачу рассеяния поля антенной системы, вектор напряженности электрического поля которой параллелен осям цилиндрических объектов. Пусть оси объектов параллельны оси z, а сами объекты представляют собой идеально проводящие поверхности. Если эти поверхности удовлетворяют условию непрерывности Гельдера [4] – в любых точках p, q принадлежащих поверхности рассеивающего препятствия S, существуют векторы нормалей $\vec{n}_p, \vec{n}_q$, для которых выполняется неравенство

$$\left|\vec{n}_p - \vec{n}_q\right| \le c\left|p - q\right|^{\alpha}, \tag{7}$$

где $c, 0 < \alpha \le 1$ – произвольные числа. В рассматриваемом варианте это условие должно быть трансформировано для контуров, образованных сечением объектов плоскостью, перпендикулярной оси z. Тогда для нахождения распределения поверхностных токов на системе N контуров $L_m, m = \overline{1,N}$, можно использовать ИУ I рода [3]:

$$E_z(x,y) = -\frac{i}{4}\sum_{m=1}^{N}\int_{L_m} j_{zm}(x',y')H_0^2(kr)dl', \tag{8}$$

где $E_z(x,y)$ – скалярная компонента напряженности электрического поля, определяемая амплитудно-фазовым распределением (АФР) возбуждения проектируемой антенной системой; $r = \sqrt{(x-x')^2 + (y-y')^2}$; $H_0^2(kr)$ – функция Ганкеля второго рода нулевого порядка. Численное решение уравнения (8) дает возможность прогнозировать поведение (АФР) токов тех объектов, влияние которых может приводить к наибольшим искажениям поля излучения. Уменьшить такое влияние можно, например, изменением расположения проектируемой антенной системы относительно цилиндрических объектов. В частности, учет влияния поясов и опор башни (мачты) в общей структуре поля излучения антенной системы дает возможность синтеза АФР токов, формирующих ДН в наибольшей степени приближенной к заданной ДН.

Численное исследование ММ рассматриваемой задачи может быть получено как прямыми, так и итерационными методами, которые имеют свои особенности и области применения. Для задач большой размерности широкое применение получили итерационные методы, например, минимизации обобщенных невязок, Ньютона – Гаусса. Используемый программный комплекс MATLAB предоставляет разработчикам ММ широкие возможности для составления программ с использованием эффективного представления матричных форм *Function Handle*, а рациональная организация блочной структуры СЛАУ с использованием неявных вложенных циклов дает возможность значительно повысить быстродействие программ. Важным достоинством итерационных методов является возможность пошагового контроля скорости сходимости и нормы невязки решения. Кроме того, предварительные численные эксперименты для однотипных задач могут дать практический интервал значений нормы невязки, при которых достигаются как удовлетворительная точность решения, так и приемлемое время расчетов.

Значительно расширяет возможности математического моделирования обратных задач рассеяния использование суперкомпьютеров. Большое количество вычислительных экспериментов, проведенных на кластере «СКИФ-Урал» Южно-Уральского государственного университета, показывают, что полный цикл моделирования электродинамических задач, включающий в себя формирование СЛАУ размерностью порядка $1{,}3\cdot 10^9$ элементов и решение прямым методом исключения Гаусса не превышает 3 минут. Перспективным направлением здесь является использование итерационных методов, позволяющих увеличить размерность задачи в десятки раз.

В отличие от гиперсингулярного ИУ Поклингтона (2) ИУ (8) относится к классу уравнений с интегрируемой логарифмической особенностью при совпадении координат точек наблюдения и интегрирования. Асимптотическое поведение функции Ганкеля при $r \to 0$ приводит к следующему представлению ядра ИУ (8):

$$H_0^{(2)}(kr)\Big|_{r\to 0} \approx 1 - i\frac{2}{\pi}\ln\frac{k\gamma\Delta l}{4e}, \qquad (9)$$

где $\gamma = 0{,}5772...$ – постоянная Эйлера; Δl – интервал дискретизации контура L_m при выделении особенности. Подчеркнем, что здесь, как и в первом примере, вводится только лишь асимптотическая оценка ядра ИУ, оторванная от оценки поведения ядра для всех остальных интервалов, где возможно применение квадратурных формул с четко определенными оценками погрешности вычислений. Для устранения такой неопределенности для вычисления коэффициентов блочной СЛАУ будем для всех интервалов дискретизации контуров L_m использовать одну и ту же квадратурную формулу повышенной точности Гаусса – Лежандра с четным количеством узлов. В этом случае автоматически блокируется возможность вычислений NaN, количество узлов можно выбирать с помощью следующей адаптивной процедуры. Выбирается начальное количество узлов N_q, проводится численное исследование ММ. Далее количество узлов последовательно увеличивается на 2, новые результаты сравниваются с полученными на предыдущем шаге для проверки критерия заданной относительной погрешности вычислений. Большое количество вычислительных экспериментов показывает, что для количества узлов $N_q = 16, 18, ..., 32$ предложенная вычислительная схема работает на 15...18% быстрее, чем традиционный подход с выделением особенности (9). Сравнение производилось для одного и того же уровня относительной погрешности при решении тестовой задачи теории рассеяния электромагнитных полей – рассеяние плоской волны на идеально проводящем круговом цилиндре. Существенное сокращение времени исследования ММ рассматриваемой задачи обусловлено однородной структурой программного комплекса вычисления коэффициентов блочной СЛАУ и рациональной организацией формирования блоков СЛАУ. На рис. 2 приведены примеры расчета двух ДН электрического диполя, расположенного вблизи идеально проводящего эллиптического цилиндра (сплошные линии), большая ось эллипса параллельна оси x.

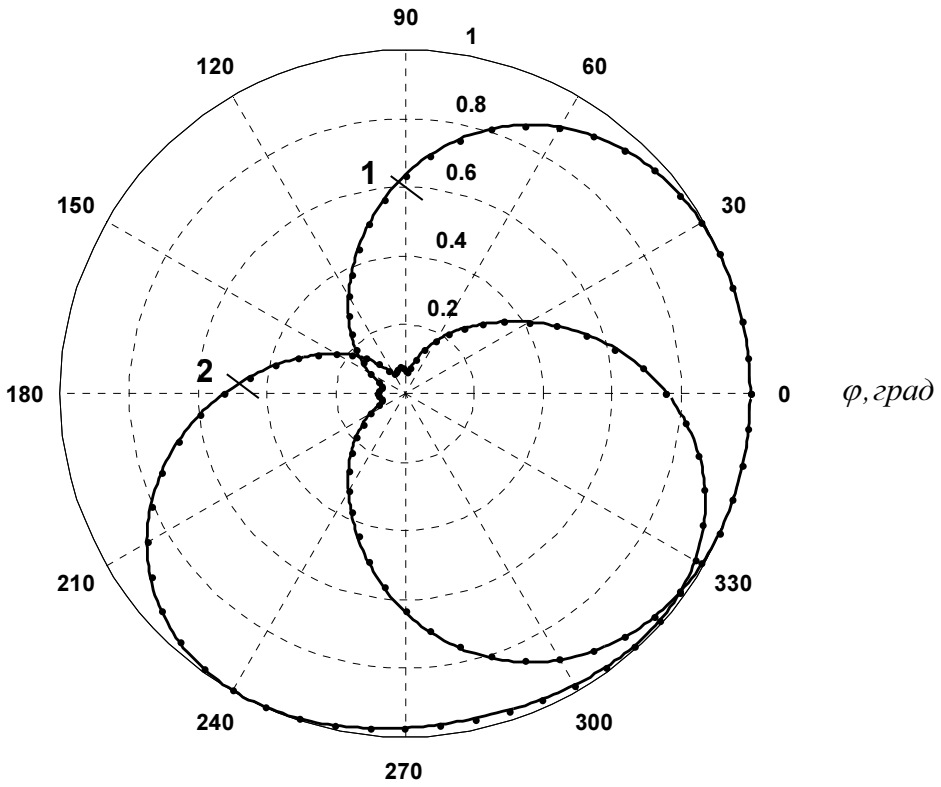


Рис. 2. ДН электрического диполя вблизи эллиптического цилиндра

Расстояние между фокусами эллипса $0,65\lambda$, эксцентриситет эллипса равен 0,72; координаты расположения диполя $(x_i, y_i), i = 1,2$. ДН 1 соответствует координатам $x_1 = 0,67\lambda;\ \ y_1 = 0$, для ДН 2 $x_2 = 0,13\lambda;\ \ y_2 = -0,61\lambda$. Точками на соответствующих ДН приведены результаты расчета ДН по аналитическим представлениям в виде рядов по функциям эллиптического цилиндра Матье, Матье-Бесселя, Матье-Ганкеля. Сравнение этих ДН показывает четкое соответствие результатов исследования ММ антенны вблизи цилиндрического объекта и известных тестовых расчетов.

3. Оценка эффективности ММ панельного излучателя с рефлектором конечной толщины

Для получения ДН, максимально приближенной к заданной ДН, и обеспечению требуемого КНД, в задачах проектирования антенных систем очень часто используются панельные излучатели. Панельные излучатели представляют собой полуволновые симметричные вибраторные антенны, расположенные над панелью-рефлектором, который из соображений минимизации массо-габаритных характеристик и уменьшения ветровой нагрузки обычно проектируется решетчатым. При этом ширина и частота расположения полос выбирается из условий необходимого коэффициента экранирования электромагнитного поля в теневой области рефлектора.

Разработка ММ такого излучателя опять приводит к необходимости использования строгой электродинамической постановки задачи формирования общего поля излучения

антенной системы, так как приближенные методы, основанные на приближении бесконечного идеально проводящего экрана, дают достаточно большие ошибки определения ДН. Необходимо подчеркнуть, что ограниченные размеры панельного излучателя приводят к тому, что вследствие дифракции электромагнитного поля вибраторных излучателей уровень поля излучения в теневой области (за панельным излучателем) может превышать заданный уровень ДН. Кроме того, характерной особенностью панельных излучателей является наличие металлической поверхности конечной толщины t, в некоторых случаях достаточно малой в сравнении с длиной волны λ. В первом приближении для математического моделирования поля излучения панельного излучателя можно считать, что поверхность панели-рефлектора является сплошной. В работе [7] показано, что переход к обоснованной геометрии решетчатой панели-рефлектора не приводит к заметному расхождению результатов, но существенно уменьшает временные затраты математического моделирования. С другой стороны, в ряде случаев толщина металлической поверхности может принимать большие значения, например, для обеспечения заданной механической прочности конструкции антенны. В связи с этим необходимо использование унифицированного подхода к построению ММ рассматриваемых антенн, общего для металлических поверхностей разной толщины t и произвольной геометрии. Кроме того, ММ исследуемых антенн должны учитывать влияние металлических поверхностей сложной формы на характеристики согласования в заданной полосе частот, поляризационные свойства поля излучения, параметры электромагнитной совместимости.

Рассмотрим метод построения ММ проектируемого панельного излучателя с использованием ИУ Фредгольма II рода, у которых ядро имеет интегрируемую слабую особенность. Вместе с тем, использование ИУ II рода затруднительно для очень малых значений t, особенно для плоских пластин. Это связано с тем, что для определенного расположения точек наблюдения и источников ядро ИУ стремится к нулю, что требует очень высокой точности вычисления подынтегральных функций для выбранного метода перехода к дискретной схеме. В связи с этим, большой практический интерес представляет исследование предельной величины t, при которой численное решение ИУ II рода для тонких металлических пластин произвольной геометрии соответствует заданным требованиям к точности и устойчивости решения для трехмерных задач. В качестве примера введем систему ИУ II рода для вибраторного излучателя:

$$\begin{aligned} &\vec{j}_s = \frac{1}{2\pi}\vec{n}_p \times \int\limits_S \vec{j}_s \times grad'\varphi\, ds' - \frac{1}{2\pi}\vec{n}_p \times \int\limits_{-h}^{h} I_x \vec{i}_x \times grad'\varphi\, dx'; \\ &-\frac{4\pi i k}{Z_c}\int\limits_S \left(j_{sx}K_{x'p} + j_{sy}K_{y'p} + j_{sz}K_{z'p}\right) ds' + \int\limits_{-h}^{h} I_x K_{x'x}\, dx' = -\frac{4\pi i k}{Z_c} E_x^i, \end{aligned} \tag{10}$$

где $\vec{j}_s = \vec{i}_x j_{sx} + \vec{i}_y j_{sy} + \vec{i}_z j_{sz}$ – распределение поверхностного тока на рефлекторе; $\varphi = \exp(-ikr)/r$; $r = \sqrt{(x-x')^2 + (y-y')^2 + t^2}$; вибратор параллелен оси x; I_x – АФР тока на вибраторе, которое определяется заданным возбуждением E_x^i и решением ИУ Поклингтона – второе слагаемое второго уравнения системы (10). В первом уравнении системы (10) точка наблюдения располагается на поверхности рефлектора, во втором уравнении – на вибраторе, при этом АФР тока I_x учитывает влияние поля рассеяния от

поверхности рефлектора на возбуждение вибратора. Это дает возможность анализировать характеристики согласования антенны в заданной полосе частот. Конкретный вид компонент ядра ИУ (10) определяется строгими электродинамическими соотношениями для векторов напряженности электрического и магнитного поля.

Численное решение системы (10) позволяет определить АФР как торцевых токов рефлектора, так и АФР токов на «теневой» поверхности, что значительно усложняет вычислительные процедуры анализа ММ антенны, но такое решение позволяет оценить тонкие эффекты взаимодействия в исследуемой антенне. Это особенно важно, если панельная антенна входит в состав антенной решетки. Определение минимального значения t/λ, при котором не нарушаются физически достоверные результаты моделирования, можно получить с помощью процедуры динамической стабилизации. Для этого выбирается начальное значение параметра t/λ, находится численное решение системы (10). Отметим, что для вычисления матричных коэффициентов, следующих из выбранного метода дискретизации задачи, здесь требуется использование квадратурных формул повышенной точности, вследствие особенностей поведения ядра ИУ. Вычислительные эксперименты показывают, что приемлемые результаты можно получить с помощью квадратурных формул Гаусса – Лежандра с числом узлов от 16 до 32, в зависимости от величины параметра t/λ. Затем производится пошаговое уменьшение параметра t/λ с одновременным контролем устойчивости численного решения, до резкого изменения АФР поверхностных токов рефлектора. Численные эксперименты показывают, что такой пошаговый процесс является достаточно монотонным и нарушение стабильного изменения токовых распределений начинаются со значений $t/\lambda < 0{,}005$. Например, для диапазона 880 – 1100 МГц минимальная толщина рефлектора равна 1,3...1,7 мм, что вполне соответствует параметрам практических конструкций панельных антенн. Таким образом, предложенная вычислительная схема повышенной точности совместно с рациональной организацией формирования блочной структуры СЛАУ для трехмерной задачи на основе вложенных неявных циклов дает возможность исследования широкого класса ММ проектируемой антенной системы. Полученные результаты хорошо согласуются с известными экспериментальными исследованиями.

4. Коррекция амплитудной ДН ФАР в заданном секторе бокового излучения

При пректировании антенных систем часто приходится учитывать ряд дополнительных ограничений, вызванных требованиями электромагнитной совместимости, особенно при расположении антенной системы на ограниченном объекте малых электрических размеров. Для компенсации влияния помеховых полей рассеяния, вызванных влиянием близкорасположенных объектов, соседних антенных систем, в амплитудной ДН необходимо минимизировать уровень боковых лепестков в заданном угловом секторе, в пределах которого проектируемая антенная система может принимать помеховые электромагнитные поля. Наиболее эффективно такая задача

может быть решена для ФАР, так как для таких антенных систем существует возможность адаптивно управлять фазовым распределением (ФР) с помощью аналоговых или дискретных фазовращателей. Применение дискретных диодных фазовращателей обеспечивает очень высокий темп переключений, при котором каждый такт соответствует своей ДН. Но при этом дискретный характер ФР не вполне соответствует оптимальному распределению, синтез которого может обеспечить наиболее точное воспроизведение требуемой ДН с минимальным уровнем в заданном угловом секторе. Амплитудное распределение (АР) возбуждения ФАР в большинстве случаев считается постоянным, так как формирование динамических АР является сложной практической задачей.

Рассмотрим следующую задачу. В рамках рассмотренных выше предположений о соответствии асимптотических решений двумерных и трехмерных задач представим ММ ФАР в виде системы $N=32$ нитей электрического тока, параллельных оси z, образующих линейную антенную решетку вертикальной поляризации над идеально проводящим плоским рефлектором. В качестве исходного АФР возбуждения ФАР используем типовой вариант спадающето к краям АР и синфазное ФР. Расстояние между излучателями для сканирующей ФАР $d=0,65\lambda$, длина рефлектора $L=20,94\lambda$, толщина t выбирается аналогично изложенной выше адаптивной процедуре, при которой обеспечивается устойчивое численное исследование ММ. Введем АФР токов в виде $j_{z,n}^{cm}(x,y,z)=I_n\delta(x-x_n)\delta(y-y_n)$, $n=\overline{1,N}$. При построении ММ задачи необходимо учитывать, что комплексная амплитуда $I_n(p_n)$ определяется как сумма двух функций:

$$I_n(p_n)=I_n^{cm}(p_n)+I_n^s(p_n),\ \ n=\overline{1,N}, \tag{11}$$

где p_n – координата расположения излучателя; $I_n^{cm}(p_n)$ – собственное АФР излучателя с номером n, создаваемое внешним (сторонним) источником; $I_n^s(p_n)$ – наведенная часть АФР, источником которой является поле рассеяния и поля других излучателей (взаимное влияние излучателей).

Введем скалярные функции $u_{i,s}(x,y)$ – продольные компоненты поля излучения антенной системы $\vec{E}_i$ и поля рассеяния $\vec{E}_s$, (E-поляризация):

$$\vec{E}=\{0,0,u(x,y)\};\ \ \vec{H}=\left\{-\frac{i}{\omega\mu}\frac{\partial u(x,y)}{\partial y},\frac{i}{\omega\mu}\frac{\partial u(x,y)}{\partial x},0\right\}, \tag{12}$$

где ω – циклическая частота электромагнитного поля. Общее поле излучения антенной системы определим следующим образом:

$$u_i(p)=\sum_{n=1}^{N}\left[I_n^{cm}+I_n^s\right]G(p,q_n);\ \ G(p,q_n)=-iH_0^2\left(kr_{pq_n}\right)\Big/4\,. \tag{13}$$

Аналогично введем поле рассеяния, источниками которого являются распределения токов на контуре рефлектора L:

$$u_s(p)=\int_L j_s(q)\,G(p,q)\,dl_q\,. \tag{14}$$

Последовательно применяя к области построения решения вторую формулу Грина с учетом соответствующих граничных условий для полного поля $u = u_i + u_s$ на контуре L и свойств непрерывности потенциала простого слоя, получим следующую систему сумматорно-интегральных уравнений I рода:

$$\begin{aligned} &\sum_{n=1}^{N} I_n^s G(p,q_n) + \int_L j_s(q) G(p,q) dl_q = 0;\ p \in p_n;\ n = \overline{1,N};\\ &\sum_{n=1}^{N} I_n^s G(p,q_n) + \int_L j_s(q) G(p,q) dl_q = -\sum_{n=1}^{N} I_n^{cm} G(p,q_n);\ p \in L. \end{aligned} \tag{15}$$

Функция Грина в (15) имеет логарифмическую особенность, поэтому все условия существования и непрерывности решения удовлетворяют требованиям для интегральных уравнений Фредгольма I рода. Таким образом, аналогично рассмотренным выше примерам, на первый план выходят вопросы достижения заданной точности вычислений при минимальных затратах компьютерного времени. После решения системы (15) асимптотическая оценка поля излучения антенной системы в дальней зоне, и в плоскости, перпендикулярной оси z, определяется следующим образом:

$$\begin{aligned} u = \sqrt{\frac{2}{\pi k r}} \exp(-i\pi/4) \Bigg[&\sum_{n=1}^{N} \left(I_n^{cm} + I_n^s \right) \exp\left(-ik\left[x_n \cos\varphi + y_n \sin\varphi\right]\right) + \\ &+ \int_L j_s(q) \exp\left(-ik\left[x_q \cos\varphi + y_q \sin\varphi\right]\right) dl_q \Bigg], \end{aligned} \tag{16}$$

где φ – азимутальный угол расчета ДН. Коррекция ФР производится на основании минимизации целевой функции следующего вида [8]:

$$\Phi(\psi) = \sum_{\varphi \in \Phi_{гл}} \left| F(\varphi) - F^* \right|^2 + w \sum_{\varphi \in \Phi_{под}} \left[F(\varphi) - F^* \right]^+ \to \min, \tag{17}$$

где вектор $\psi = \arg I^{cm}$ – фазовое распределение возбуждения ФАР; $F(\varphi)$ – нормированная амплитудная ДН ФАР, определяемая из (16); F^* – требуемая амплитудная ДН с подавлением бокового излучения в секторе азимутальных углов $\Phi_{под}$; $\Phi_{гл}$ – сектор углов, занимаемый главным лепестком ДН; w – адаптивный весовой коэффициент, регулирующий вклад в целевую функцию отклонений только в области главного лепестка, символ $[\ ]^+$ означает суммирование только положительных отклонений от заданного уровня бокового излучения в секторе $\Phi_{под}$.

Следовательно, на каждом шаге процедуры минимизации целевой функции $\Phi(\psi)$ производится решение системы уравнений (15), поэтому повышенные требования к точности и скорости вычислений для рассматриваемой задачи имеют принципиальное значение. Вычислительные эксперименты показывают, что приемлемые результаты можно получить с помощью квадратурных формул Гаусса – Лежандра с числом узлов от 16 до 32, в зависимости от величины параметра t/λ. Для минимизации целевой функции $\Phi(\psi)$ используются программы *fminsearch* и *fminunc* системы MATLAB.

На рис. 3 приведены результаты коррекции нормированной амплитудной ДН с подавлением бокового излучения в секторе углов $4{,}2^{\circ} < \varphi < 10^{\circ}$.

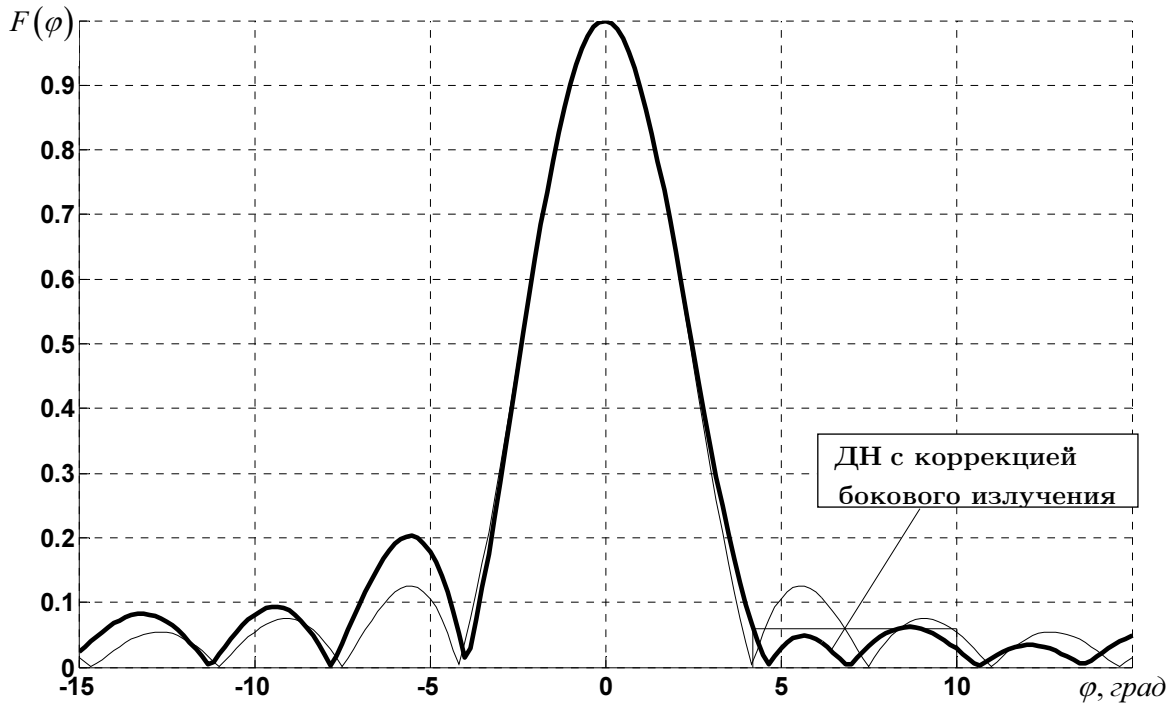


Рис. 3. Коррекция бокового излучения в заданном секторе углов

Уровень бокового излучения в этом секторе не должен превышать значение 0,07. Для управления ФР используем фазовращатели с дискретом фазирования $11{,}25^{\circ}$. АР примем фиксированным: излучатели с 1 по 8 и с 25 по 32 имеют уровень нормированной амплитуды 0,5; с 9 по 24 – уровень 1. Такое АР легко реализуется бинарными схемами деления на основе квадратурных или синфазных мостов. Исходная ДН (тонкие линии) соответствует заданному АР и синфазному ФР. Отметим, что любые асимметричные изменения ФР, возникающие при работе программ минимизации целевой функции, неизбежно ведут к появлению негативных факторов – отклонение главного лепестка ДН и его расширению. Для уменьшения этих факторов используется весовой коэффициент w, который можно определить после проведения ряда вычислительных экспериментов. Приведенные результаты показывают, что оптимальное дискретное ФР ведет к снижению уровня бокового излучения в заданном секторе углов при устойчивой фиксации главного лепестка. Но при этом обязательно наблюдается увеличение уровня бокового излучения с противоположной стороны главного лепестка ДН, поэтому общая процедура минимизации целевой функции должна контролировать полную картину бокового излучения для оперативной коррекции фазового распределения.

Заключение

1. Использование квадратурных формул Гаусс – Лежандра повышенной точности позволяет исследовать ММ проектируемых антенных систем без проведения процедур выделения особенностей ИУ, которые дают только асимптотические оценки поведения ядра ИУ. Прямое вычисление коэффициентов СЛАУ с использованием эффективных программ численного интегрирования дает возможность получения оценок

погрешностей для однородного характера вычислительных процедур. При этом отмечается заметное уменьшение времени вычислений, что представляет практический интерес при проведении очень большого количества циклических расчетов ДН, особенно при решении СЛАУ высоких порядков, проведении оптимизации поля излучения антенной системы.

2. Рациональная организация формирования блочной структуры СЛАУ, к которым редуцируются ИУ, существенно ускоряет процесс заполнения матриц ММ. Это расширяет возможности математического моделирования сложных антенных систем. Сравнение полученных результатов с известными тестовыми решениями показывает высокую эффективность и универсальность предложенных вычислительных схем повышенной точности.

Литература

1. Ильинский, А.С. Математические модели электродинамики / А.С. Ильинский, В.В. Кравцов, А.Г. А.Г. Свешников — М: Высшая школа, 1991. — 224 с.
2. Хижняк, Н.А. Интегральные уравнения макроскопической электродинамики. — Киев: Наукова думка, 1986. — 280 с.
3. Вычислительные методы в электродинамике / Под ред. Р. Митры. М.: Мир, 1977. — 488 с.
4. Колтон, Д. Методы интегральных уравнений в теории рассеяния / Д. Колтон, Р. Кресс. — М.: Мир, 1987. — 312 с.
5. S Sophocles J. Orfanidis. Electromagnetic Waves and Antennas. — 2008. ˜ www.ece.rutgers.edu/˜orfanidi/ewa.
6. Войтович, Н.И. О соответствии асимптотических решений двумерных и трехмерных задач в антенной технике / Н. И. Войтович, А. Б. Хашимов // Радиотехника и электроника. — 2010. — Т.55, № 12. — С. 1471-1476.
7. Воробьев, М.С. Особенности диаграммы направленности антенной системы, расположенной на башне / М.С. Воробьев, Л. П. Кудрин, А.П. Медведев, А.Б. Хашимов // Электросвязь. — 2014. — № 1. — С. 23–29.
8. Малакшинов, Н.П. Численное решение некоторых задач прикладной электродинамики с применением методов аппроксимации и оптимизации / Н.П. Малакшинов Н.П., В.Г. Ерихов, В.Н. Гармаш, А.Н. Егоров // Сборник научно-методических статей по прикладной электродинамике. — М.: Высшая школа, 1980. — Вып. 4. — С. 68–94.

Хашимов Амур Бариевич, канд.физ.-мат. наук, доцент кафедры конструирования и производства радиоаппаратуры, Южно-Уральский государственный университет (г. Челябинск); xab@kipr.susu.ac.ru.

Салихов Ринат Рафикович, главный конструктор ООО «НПО РТС», (г. Челябинск); sal_rr@mail.ru.

Альметов Руслан Салаватович, студент кафедры конструирования и производства радиоаппаратуры, Южно-Уральский государственный университет (г. Челябинск); ruslannss@mail.ru.

Поступила в редакцию 9 апреля 2014 г.

Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 2, pp. 77–91

USING OF THE HIGH PRECISION NUMERICAL TECHNIQUES FOR ANTENNA SYSTEM DESIGN

A.B. Khashimov, South Ural State University, Chelyabinsk, Russian Federation, xab@kipr.susu.ac.ru.

R.R. Salikhov, JSC «NPO RTS», Chelyabinsk, Russian Federation, sal_rr@mail.ru.

R.S. Al'metov, South Ural State University, Chelyabinsk, Russian Federation.

This paper proposes the some effective numerical methods for studing mathematical models of antenna system.

This paper proposes the some effective numerical methods for studing mathematical models of antenna system. Formulation of mathematical models is used the rigorous electromagnetic principies and equations. Using the high precision quadrature formulas for solving Pocklington's integral equation and special regularizing procedure is provided the stable modeling results with given accuracy and high speed.

Keywords: mathematical model, integral equations, quadrature formulas of Gauss-Legendre.

References

1. Il'insky A.S., Kravtsov V.V., Sveshnikov A.G. Matematicheskie modeli elektrodinamiki [Mathematical Models of the Eelectomagnetics],Moscow, Vysshaya Shkola, 1991, 224 p.
2. Khizhnyak N.A. Integralnyie uravneniya makroskopicheskoi elektrodinamiki [Integral Equations of Electromagnetics], Kiev, Naukova dumka, 1986, 280 p.
3. Computing Techniques for Electromagnetics. Edited by R. Mittra. Pergamon Press. Oxford, New York, Toronto, Sydney, Braunschweig. 1973. – 488 p.
4. Colton D., Kress R. Integral Equation Method in Scattering Theory. John Wiley & Sons. New York, Chichester, Brisbane, Toronto, Singapore. 1983. – 312 p.
5. Sophocles J. Orfanidis. Electromagnetic Waves and Antennas. – 2008. ~ www.ece.rutgers.edu/~orfanidi/ewa.
6. Voitivich N.I., Khashimov A.B. On the Correspondence of Asymptotic Solutions to 2D and 3D Problems in Antenna Engineering, Journal of Communications Technology and Electronics, 2010, Vol. 55, No. 12, pp. 1374–1379. Pleiades Publishing, Inc., 2010.
7. Vorob'ev M.S., Kudrin L.P., Medvedev A.P., Khashimov A.B. Osobennosti diagrammy napravlennosti antennoi sistemy, raspolozhennoi na bashne [Radition Pattern Features of the Antenna System Disposed on a Tower] // Elektrosvyaz. – 2014. – № 1. – pp. 23–29.
8. Malakshinov N.P., Erikhov V.G., Garmash V.N., Egorov A.N. Chislennoe reshenie nekotoryh zadach prikladnoi elektrodinamiki c primeneniem metodov approksimatsii i optimizatsii // Sbornik nauchno-metodicheskih statei po prikladnoi elektrodinamike. – Moscow, Vysshaya Shkola, 1980. – pp. 68–94.

Received 9 April 2014.

Краткие сообщения

УДК 519.612, 534.134

ИССЛЕДОВАНИЕ ГАЗОДИНАМИЧЕСКОГО ПОТОКА И КОНСТРУКЦИИ В МОДЕЛЬНОЙ ЭКСПЕРИМЕНТАЛЬНОЙ УСТАНОВКЕ[1]

Л.Н. Бутымова, В.Я. Модорский

В данной работе проводится анализ влияния распараллеливания задачи и параметров расчетной сетки на время выполнения задачи по расчету газодинамического потока и напряженно-деформированного состояния конструкции рабочей камеры экспериментальной установки. Разработаны физическая, математическая, твердотельная и сеточная модели отдельно для газодинамической задачи и задачи теории упругости. Обсуждаются некоторые полученные результаты вычислительных экспериментов.Обнаружено, что масштабируемость прочностного расчета ниже, чем при оценке газодинамического процесса. Предварительный анализ показал, что скорость расчета газодинамических процессов повышается при оптимизации процессов распараллеливания и построения сетки. Масштабируемость по оценке напряженно-деформированного состояния конструкции при этом не повышается. При решении связанных задач, применительно к динамическим системам «газ-конструкция» в системе инженерного анализа Ansys возможно, что продолжительность расчета окажется неприемлемой. Для решения такого рода задач, возможно, потребуется разработка новых алгоритмов.

Ключевые слова: газодинамический поток, напряженно-деформированное состояние конструкции, экспериментальная установка, вычислительный эксперимент, вибрации, масштабируемость, параметры расчетной сетки, распараллеливание.

Введение

Аэроупругие явления могут приводить к отказам в работе энергетических установок ракет, самолетов, к разрушению строительных конструкций, трубопроводов различного назначения и т. д. Опасность аэроупругих явлений приводит к непрогнозируемым последствиям, которые можно обнаружить только на этапе создания натурного образца. Многовариантность конкретных реализаций аэроупругих процессов, а также сама сложность эти явлений предполагает необходимость прогноза, который может быть осуществлен с помощью вычислительного эксперимента. Для повышения надежности прогнозирования необходима верификация расчетных методик, но с использованием, по возможности физического модельного эксперимента. В то же время недостаточно существующих расчетно-экспериментальных методик. Данная статья описывает процесс создания и использования расчетно-экспериментальных методик оценки колебательных процессов в аэроупругой постановке с использованием Супер ЭВМ.

Цель работы: создание и отработка методики исследования аэроупругих процессов на базе расчетно-экспериментального комплекса с использованием Супер ЭВМ.

[1]Статья рекомендована к публикации программным комитетом Международной научной конференции "Параллельные вычислительные технологии – 2014".

Большинство методик исследования аэроупругих явлений используют одно- и двухмерные модели, чаще в стационарной и квазистационарной линейной постановке, и базируются на использовании волнового уравнения. В целом принимается, что развитие колебаний зависит от относительного вклада различных физико-химических процессов, способствующих усилению, либо демпфированию возмущений. Обычно принимается, чтогаз воздействует на конструкцию, но необходим учет и их взаимовлияния. Кроме того, имеющиеся методики не могут объяснить влияние жесткостных и массовых характеристик, условий закрепления конструкции на колебания давления в ЭУ [1].Особый интерес представляет поиск механизмов возникновения быстропротекающих колебательных процессов, характерное время которых имеет длительность, соизмеримую со временем распространения возмущения в конструкции.

1. Теоретическая часть

1.1. Газодинамический расчет

Сформулирована следующая физическая постановка задачи:

– процессы рассматриваются трехмерными динамическими; в качестве рабочего тела выбран воздух, который рассматривается как сжимаемый; стенки модельной камеры непроницаемые, нетеплопроводные; в начальный момент времени модельная камера заполнена воздухом, значение давления для первого варианта равно 0,3 МПа; для второго варианта – 0,1 МПа, температура воздуха – 293 К [2].

Математическая модель движения газа – это система уравнений в частных производных, включающая в себя законы сохранения массы, импульса, энергии и уравнение состояниягаза. Математическая модель может быть модифицирована через интерфейс Ansys CFX.

Рассчитываемыми переменными математической модели являются векторные и скалярные поля, разделяемые на независимые и зависимые переменные. Независимые переменные определяются из решения систем уравнений с совокупностью начальных и граничных условий, а зависимые выражаются через них.

Выбранная математическая модель включает в себя следующие соотношения:

– закон сохранения массы

$$\frac{\partial \rho}{\partial t} + \nabla\left(\rho \vec{V}\right) = 0$$

– закон сохранения импульса

$$\frac{\partial \rho \vec{V}}{\partial t} + \nabla\left(\rho \vec{V} \otimes \vec{V}\right) = -\nabla \mathrm{P} + \nabla\left((\mu + \mu_t)\left(\nabla \vec{V} + \left(\nabla \vec{V}\right)^T\right)\right)$$

– закон сохранения энергии

$$\frac{\partial (\rho H)}{\partial t} + \nabla\left(\rho \vec{V} H\right) = \frac{\partial P}{\partial t} + \nabla\left(\left(\frac{\lambda}{c_p} + \frac{\mu_t}{Pr_t}\right)\nabla \mathrm{H}\right)$$

где t – время; ρ – плотность; V – скорость; P – давление; μ – динамическая вязкость; μ_t – турбулентная динамическая вязкость; H – энтальпия; c_p – теплоемкость; λ – коэффициент теплопроводности; Pr_t – турбулентное число Прандтля.

Используется известная $k - \varepsilon$ модель турбулентности. Вводятся следующие соотношения:

$$\frac{\partial (\rho k)}{\partial t} + \nabla\left(\rho \vec{V} k\right) = \nabla\left(\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right) + \mu_t G - \rho \varepsilon$$

$$\frac{\partial(\rho\varepsilon)}{\partial t}+\nabla\left(\rho\vec{V}\varepsilon\right)=\nabla\left(\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\nabla\varepsilon\right)+C_1\frac{\epsilon}{k}\mu_t G-C_2 f_1\rho\frac{\varepsilon^2}{k}$$

где $G=D_{ij}\frac{\partial V_i}{\partial x_i}$, $D_{ij}=S_{ij}-\frac{2}{3}\left(\nabla\cdot\vec{V}+\frac{\rho k}{\mu_t}\right)\delta_{ij}$, $S_{ij}=\frac{\partial V_i}{\partial x_j}+\frac{\partial V_j}{\partial x_i}$, $\sigma_k=1{,}\sigma_\varepsilon=1{,}3$, $C_1=1{,}44{,}C_2=1{,}92$, $f_1=1$, k – турбулентная энергия, ϵ – скорость диссипации турбулентной энергии, G – определяет скорость генерации турбулентной энергии, S_{ij} – удвоенный тензор скоростей деформации.

Уравнение турбулентной вязкости вычисляется по формуле Колмогорова-Прандтля:

$$\mu_t=C_\mu\rho\frac{k^2}{\varepsilon}$$

где $C_\mu=0{,}09$ – константа.

– уравнение состояния

$$P=\rho RT$$

где R - универсальная газовая постоянная.

Математическая модель замыкается начальными и граничными условиями.

1.2. Расчет напряженно-деформированного состояния конструкции

Сформулирована следующая физическая постановка:

– конструкция рассматривается трехмерной; исследуется модельная камера, физико-механические характеристики и геометрические параметры которой, можно изменять;материал модельной камеры – упругий, однородный.

Модальный анализ служит для определения собственных частот и форм колебаний конструкции.Результаты анализа дают возможность установить число форм колебаний и шаг интегрирования по времени, что может обеспечить надежное решение задачи о динамическом поведении системы в неустановившемся режиме.

В программе Ansys Workbench модальный анализ – это решение задачи о свободных (невынужденных), затухающих или незатухающих, колебаниях дискретной системы, которая описывается следующим уравнением движения:

$$[M]\{u''\}+[C]\{u'\}+[K]\{u\}=0$$

где $[M]$ – матрица масс; $[C]$ – матрица демпфирования; $[K]$ – матрица жесткости; $\{u''\}$ – вектор узловых ускорений; $\{u'\}$ – вектор узловых скоростей; $\{u\}$ – вектор узловых перемещений.

Этому уравнению придается форма, соответствующая задаче о собственных значениях. Для случая незатухающих колебаний (наиболее типичного для модального анализа) пренебрегают слагаемым $[C]\{u'\}$, и уравнение приводится к виду

$$([K]-\omega^2[M])\{u\}=0$$

где ω^2 – (квадрат собственной частоты) – собственное значение, $\{u\}$– собственные формы колебаний, не являющиеся функциями времени.

Допущения при расчете свободных колебаний:

– матрицы $[K]$ и $[M]$ являются константами, что означает следующее:

– принимается линейное упругое поведение материалов; используется теория малых деформаций, не учитываются нелинейные эффекты; матрица $[C]$ отсутствует, значит не учитываются эффекты демпфирования; вектор сил $\{F\}$ отсутствует, значит не рассчитывается возбуждение конструкции внешними силами; закрепление конструкции не является обязательным условием расчета, как в задачах статики;форма свободных коле-

баний вычисляется в относительных единицах и не позволяет определить абсолютные смещения.

Анализ гармонического воздействия используется для определения параметров установившегося движения линейной системы при синусоидальном силовом возбуждении. Этот вид анализа актуален при исследовании влияния нагрузок, меняющихся во времени по гармоническому закону.

Разрешающее уравнение для этого вида анализа представляет собой специальный случай общего уравнения движения, в котором вынуждающая сила $\{F(t)\}$ является синусоидальной функцией времени с известной амплитудой F_0, частотой ω и фазовым углом φ:

$$\{F(t)\} = \{F_0(cos(\omega t) + \varphi) + i\, sin((\omega t) + \varphi)\}$$

Перемещения меняются синусоидально с частотой ω, но не обязаны совпадать по фазе с вынуждающей силой. Нагрузки могут быть в форме узловых сил, начальных перемещений или распределенными по элементу. Для заданной частоты пользователь имеет возможность найти перемещения либо в виде набора амплитуд и фазовых углов, либо в виде реальной и мнимой частей решения.

Для проведения гармонического анализа доступны три метода: полный, редуцированный и метод суперпозиций. Полный метод предполагает использование полных, возможно, несимметричных, матриц $[K]$, $[M]$ и $[C]$. Это делает возможным выполнение некоторых сложных видов гармонического анализа в области акустики, пьезоэлектричества и динамики вращающихся валов. Этот метод анализа является одношаговым. Метод приведения (редуцированный) и метод суперпозиции форм колебаний можно использовать для линейных задач, приводящих к симметричным матрицам. Редуцированный метод основан на сокращении матриц $[K]$, $[M]$ и $[C]$ способом приведения Гуяня и дает решение за два шага (решение приведенного уравнения движения и затем повторный проход - шаг расширения решения). Многошаговый метод суперпозиции требует предварительного проведения модального анализа для выполнения следующих шагов решения.

Математическая модель замыкается граничными условиями.

При анализе свободных колебаниймеханические и тепловые нагрузкине задаются. Можно задавать способ закрепления конструкции. Используется фиксированное закрепление модельной камеры – FixedSupport (жесткая заделка). При проведении гармонического анализаприложим циклические перемещения (Displacement)к торцу модельной камеры. Амплитударавна 1 мкм.

2. Проведение вычислительных экспериментов

При подготовке и проведении вычислительного эксперимента использовалась система инженерного анализа Ansys.

Ansys – программный пакет конечно-элементного анализа, решающий задачи в различных областях инженерной деятельности (прочность конструкций, термодинамика, динамика жидкостей и газов, электромагнетизм), включая связанный междисциплинарный анализ.

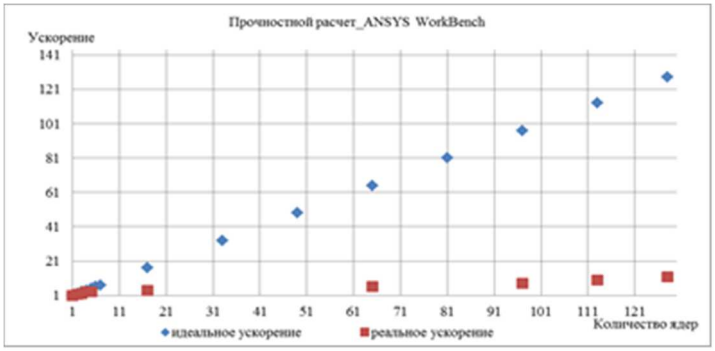
При проведении вычислительных экспериментов использовался кластер Пермского национального исследовательского политехнического университета (ПНИПУ), имеющий следующие технические характеристики:

– 88 вычислительных узлов; 128 четырехядерных процессоров «Barcelona-3» (всего 512 ядер); 48 восьмиядерных процессоров «IntelXeon E5-2680» (всего 384 ядра); пиковая производительность 20 Тфлопс; производительность в тестовом пакете Linpack 78%; объем системы хранения информации 12 ТБ; объем оперативной памяти 128 Гбайт/узел; операционная система Windows HPC Server 2008; 13-е место в рейтинге ТОП-50 в СНГ (по состоянию на 01.09.2008г.).

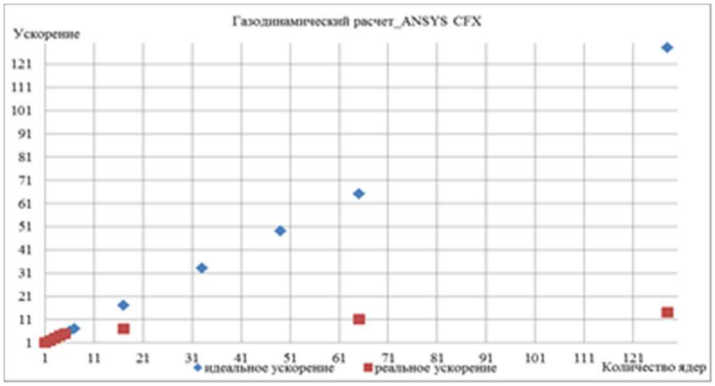
Было получено ускорение (рис. 1), рассчитанное при использовании параллельного алгоритма для p процессоров, по сравнению с последовательным вариантом выполнения вычислений, определяемое по формуле

$$S_p(n) = T_1(n)/T_p(n),$$

где величина n используется для параметризации вычислительной сложности решаемой задачи и может пониматься, например, как количество входных данных задачи, $T_1(n)$ – время последовательного решения задачи на ЭВМ, $T_p(n)$ – время выполнения алгоритма на p процессорах.



а



б

Рис. 1. Ускорение при использовании параллельного алгоритма:
а – прочностной расчет; б – газодинамический расчет

Проведена проверка сходимости решения на сгущающихся сетках.

Для верификации расчетов междисциплинарных аэроупругих задач была разработана модельная экспериментальная установка (рис.2). Установка состоит из модельной камеры, основания, генератора колебаний, системы регистрации и измерения National Instruments. В качестве чувствительных элементов используются высокочастотные вибродатчики с рабочим диапазоном частот от 0 до 100 кГц и датчики давления, работающие в диапазоне частот от 0 до 40 кГц.

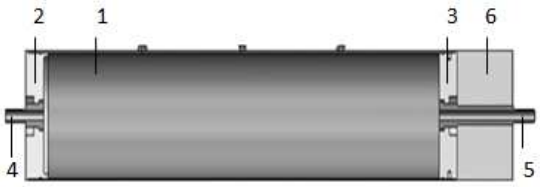


Рис. 2. Конструктивная схема экспериментальной установки:
1– корпус, 2, 3 – крышки, 4, 5 – штуцеры, 6 – сменные элементы

При этом потребовалась реализация следующих этапов проведения физического эксперимента: анализ характерных геометрических размеров исследуемого объекта, физико-механических характеристик материалов, из которых данный объект изготовлен, конструктивных особенностей, значимых с точки зрения колебательных процессов, анализ действующих нагрузок, создание виртуальной модели объекта физического эксперимента, вычислительный эксперимент по оценке собственных и вынужденных колебаний, верификация, анализ результатов вычислительного и физического экспериментов.

3. Экспериментальная часть

Для проведения экспериментальных исследований влияния физико-механических характеристик материала конструкции на характеристики резонансных колебаний необходимо было подобрать такие геометрические и массовые характеристики модельной камеры, чтобы был обеспечен резонанс модельной камеры и газовой полости.

Задавались следующие исходные данные:

Таблица 1

Исходные данные для вычислительного эксперимента

Материал	**Сталь $i = 1$**	**Титановый сплав $i = 2$**	**Алюминиевый сплав $i = 3$**
Толщина оболочки, δ, м	0,001	0,001	0,001
Длина оболочки, L, м	1	1	1

В результате расчетов определены$m^i_{\text{рез}}$:

- для стали ($i = 1$): $m^1_{\text{рез}} = 24{,}27$ кг;
- для титанового сплава ($i = 2$): $m^2_{\text{рез}} = 12{,}14$ кг;
- для алюминиевого сплава ($i = 3$): $m^3_{\text{рез}} = 8{,}5$ кг.

Вместе с тем проведены численные расчеты для труб со следующими характеристиками:

Таблица 2

Характеристики модельной камеры

Материал	**Сталь $i = 1$**	**Титановый сплав $i = 2$**	**Алюминиевый сплав $i = 3$**
Толщина оболочки, δ, м	0,001	0,001	0,001
Длина оболочки, L, м	1	1	1

В результате численных расчетов в Ansys Workbench определены $f^i_{\text{рез}}$:

для стали ($i = 1$): $f^1_{\text{рез}} = 367$ Гц;

для титанового сплава ($i = 2$): $f^2_{\text{рез}} = 333$ Гц;

для алюминиевого сплава ($i = 3$): $f^3_{\text{рез}} = 350$ Гц.

Физический эксперимент показал следующие результаты:

– для корпуса из алюминия резонансная частота для конструкции находится в пределах от 250 Гц до 350 Гц, а для газа от 200 Гц до 400 Гц;

– для корпуса излатуни резонансная частота для конструкции находится в пределах от 350 Гц до 550 Гц, а для газа от 200 Гц до 400 Гц;

– для корпуса из меди резонансная частота для конструкции находится в пределах от 700 Гц до 850 Гц, а для газа от 200 Гц до 350 Гц;

– для корпуса из стали с толщиной стенки 0,5 мм резонансная частота для конструкции находится в пределах от 700 Гц до 900 Гц, а для газа от 200 Гц до 350 Гц [3].

Заключение

1. Принят и реализован комплексный подход к вычислительному моделированию работыэкспериментальной установки: исследованы как газодинамические процессы, так и напряженно-деформированное состояние конструкции.

2. Разработаны физическая и математическая постановки и 3D модели рабочей камеры экспериментальной установки для оценки газодинамических параметров и компонент и оценке напряженно-деформированного состояния вычислительного эксперимента в системе инженерного анализа Ansys.

3. Подготовлены и проведены газодинамические и прочностные вычислительные эксперименты, в ходе которых рассмотрены переходные процессы в модельной камере для нескольких вариантов ее исполнения.

4. Проведен анализ собственных и вынужденных колебаний модельной камеры. Определены формы и частоты собственных колебаний конструкции. Определены формы, амплитуды и частоты вынужденных колебаний.

5. В ходе анализа напряженно-деформированного состояния конструкции модельной камеры получены поля перемещений, деформации и напряжений.

6. Разработана конфигурация системы измерения и регистрации экспериментальной установки с использованием оборудования фирмы National Instruments и программного обеспечения LabVIEW.

7. Подготовлены и проведены физические эксперименты, в ходе которых подтверждена работоспособность основных элементов экспериментальной установки; проведено исследование сигналов вибродатчиков, датчиков давления, установленных вдоль образующей модельной камеры; выявлено резонансное взаимодействие корпуса установки и газовой полости на частоте около 300 Гц.

8. Обнаружено, что масштабируемость прочностного расчета ниже, чем при оценке газодинамического процесса. Предварительный анализ показал, что скорость расчета газодинамических процессов повышается при оптимизации процессов распараллеливания и построения сетки. Масштабируемость по оценке напряженно-деформированного состояния конструкции при этом не повышается. При решении связанных задач, применительно к динамическим системам «газ-конструкция» в системе инженерного анализа Ansys возможно, что продолжительность расчета окажется неприемлемой. Для решения такого рода задач, возможно, потребуется разработка новых алгоритмов.

Литература

1. Модорский В.Я., Соколкин Ю.В. Газоупругие процессы в энергетических установках. / Под ред. Ю.В. Соколкина. — М.: ФИЗМАТЛИТ, 2007. — 176 с.
2. Бутымова Л.Н., Модорский В.Я., Соколкин Ю.В. Разработка экспериментальной установки и исследование влияния материала корпуса на резонансные частоты в системе «газ-конструкция» // Научно-технический вестник Поволжья. — 2013. — № 6. — С. 197–200.
3. Бутымова Л.Н., Модорский В.Я. Исследование колебательных процессов на резонансных режимах в модельной установке // Научно-технический вестник Поволжья. — 2013. — № 6. — С. 193–196.

Бутымова Лилия Николаевна, программист Центра высокопроизводительных вычислительных систем, Пермский национальный исследовательский политехнический университет (Пермь, Российская Федерация), liliya85@gmail.com.

Модорский Владимир Яковлевич, д.т.н., профессор кафедры «Механика композиционных материалов и конструкций», директор Центра высокопроизводительных вычислительных систем Пермский национальный исследовательский политехнический университет (Пермь, Российская Федерация), modorsky@pstu.ru

Поступила в редакцию 11 апреля 2014 г.

Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 2, pp. 92–100

INVESTIGATION OF GAS DYNAMIC FLOW AND STRUCTURE IN EXPERIMENTAL EQUIPMENT

L.N. Butymova, Perm National Research Polytechnic University
(Perm, Russian Federation),
V.Ya. Modorsky, Perm National Research Polytechnic University
(Perm, Russian Federation)

This paper have analyzes the influence of parallelization and computational grid parameters on execution time analysis of gas-dynamic flow and stress-strain state of the structure in the experimental equipment. It discusses some results of computational experiments. Physical, mathematical, and solid-state grid models were developed separately for gas dynamic tasks and tasks the theory of elasticity. The some results discuss of computational experiments. It is found that the strength's calculation's scalability lower than gas flow's process in the evaluation. Preliminary analysis has shown that the velocity of gas-dynamic processes of calculation raises at parallelization process optimization and constructing the mesh. Scalability is not increased the evaluation of stress-strain state of the structure. Possible that duration of the calculation will be unacceptable at the decision of related the tasks in relation to dynamical systems "gas-construction" in the system engineering analysis Ansys. For solving this kind of tasks may need to develop new algorithms.

Keywords: gas dynamic flow, stress-strain state of the structure, the experimental equipment, computational experiment, vibration, scalability, parameters of the computational grid, paralleling.

References

1. Modorsky V.Ya, Sokolkin Y.V. Gas–Elasticity processes in power plants. / Ed. Y. Sokolkin.—M.: FIZMATLIT, 2007. — S. 176.
2. Butymova L.N., Modorsky V.Ya., Sokolkin Y.V. Development the experimental equipment and research influence of the material properties on the resonant frequencies in the system «gas–construction» // Scientific and Technical Bulletin of the Volga region. — 2013. — № 6. —P. 197–200.
3. Butymova L.N., Modorsky V.Ya. Research of oscillatory processes in resonance modes in the model construction // Scientific and Technical Bulletin of the Volga region. — 2013. — № 6. —P. 193–196.

Received 11 April 2014.

УДК 519.613.2

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ АЛГОРИТМ ШЕРМАНА – МОРРИСОНА ОБРАЩЕНИЯ МАТРИЦ НА GPU [1]

Н.С. Недожогин, А.С. Сармакеева, С.П. Копысов

Обращение матрицы является важным этапом при численном решении таких, задач как решение систем линейных уравнений и построение предобуславливателей, вычисление дополнения Шура в методах декомпозиции области, цифровая обработка изображений и т. д. Разработка высопроизводительных параллельных алгоритмов обращения матриц связана с эффективным хранением и отображением алгоритмов на современные многоядерные архитектуры. Наряду с традиционными методами обращения — LU-факторизацией и методом Гаусса – Жордана, рассмотрены параллельные алгоритмы метода сопряженных градиентов и Шермана – Моррисона, в которых используются матрично-векторные и скалярные произведения эффективно выполняемые на многоядерных процессорах. В работе проведено сравнение на тестовых матрицах рассматриваемых методов на CPU и GPU.

Ключевые слова: высокопроизводительные алгоритмы, обращение матриц, разреженные матрицы, алгоритм Шермана – Моррисона.

Введение

При решении практических задач возникает необходимость выбора того или иного метода обращения матриц или его параллельной реализации. Появление программного обеспечения для вычислений общего назначения на графических устройствах (GPGPU) позволило на ряде задач, в том числе вычислительной линейной алгебры, получать ускорение вычислений в десятки и сотни раз, по сравнению с центральным процессором (CPU). Тысячи потоков GPU могут эффективно выполнять одновременно большое число простых арифметических операций, что характерно для мультипликативных и аддитивных операций с векторами и матрицами. Вместе с тем, последовательные операции и ветвления, характерные для разложения матриц на треугольные множители, выполняются GPU медленнее, чем ядрами CPU.

Данная работа ориентирована на эффективные реализации методов обращения симметричных разрежённых и положительно определенных матриц с использованием гибридных вычислительных узлов с GPGPU. В работе проведено сравнение алгоритмов обращения матриц, имеющих разреженную структуру. При реализации всех методов учитывалась структура хранения в сжатом формате.

1. Алгоритмы обращения матриц

Рассмотрим высокопроизводительные алгоритмы явного и неявного обращения матриц на гибридной архитектуре с выделением операций эффективно выполняемых на универсальных и графических процессорах. Хотя явное обращение матрицы во многих случаях может быть нецелесообразным, оно по-прежнему представляет интерес в различных прикладных областях, где встречаются большие разреженные и, порой, плохообусловленные матрицы.

[1]Статья рекомендована к публикации программным комитетом научной конференции «Параллельные вычислительные технологии — 2014».

LU-факторизация (LU). Для заданной невырожденной матрицы A существует разложение на верхнюю треугольную U и на нижнюю треугольную L матрицы. Если матрицы A, U, L обратимы, то справедливо следующее утверждение: $A^{-1} = U^{-1}L^{-1}$. Так как факторизация проводилась для матриц хранящихся в формате CSR, при построении L и U суммировались лишь те произведения, в которых участвовали ненулевые элементы исходной матрицы [1]. Рассматривалась только последовательная реализация LU-факторизации на CPU, показывающая наилучшие результаты для малых и средних матриц. Особенности данного алгоритма не позволяют эффективно использовать все возможности массивно-параллельной архитектуры GPU.

Метод Гаусса – Жордана (GJ). Потенциально, метод Гаусса – Жордана больше подходит для вычислений на графических ускорителях, чем алгоритм основанный на LU-разложении. В рассматриваемом варианте все операции выполняются на центральном процессоре CPU. Так как преобладающие вычисления в данном методе содержат матричные операции, то в дальнейшем можно ожидать высокую производительность варианта для гибридной архитектуры при совместном использовании CPU+GPU. Однако, в работе [2] показано, что максимально получаемое трехкратное ускорение метода обращения Гаусса – Жордана достигается лишь при вычислениях с одинарной точностью и разделением операций выполняемых на CPU и GPU.

Метод сопряженных градиентов (CG). Рассматриваемый в работе алгоритм вычисления обратной матрицы состоит из решений матричной системы вида $AX = E$, где E — единичная матрица. Такая система эффективно решается на GPU предобусловленным алгоритмом сопряженных градиентов. Использование в тестах простейшего диагонального предобуславливателя связано с обеспечением минимальных вычислительных затрат (количества требуемой памяти, времени построения предобуславливателя и времени его решения) на GPU для матриц общего вида.

Реализация алгоритма более подробно рассмотрена в работе [3]. Отметим только, что оценка числа ненулевых элементов обратной матрицы затруднена и обращенную матрицу приходится хранить полностью, что накладывает существенные ограничения на используемую память при работе с GPU. В случае обращения матрицы методом сопряженных градиентов результат вычислений получается по столбцам, что позволяет сразу преобразовывать матрицу в сжатый формат и, тем самым, снижать ограничение на размер задачи, решаемой на GPU. В методе сопряженных градиентов операции скалярного произведения, нормы векторов, суммы векторов и умножения вектора на скаляр распараллелены с помощью библиотеки cuBLAS. Матрично-векторное произведение реализовано с помощью технологии CUDA в векторном варианте с возможностью использования нескольких GPU.

Метод Шермана – Моррисона (SM). Идея метода SM восходит к работам [4, 5], который известен в русскоязычной литературе метод как метод пополнения [6]. Сначала рассмотрим тождество Вудбери [4] из которого следует известная формула Шермана – Моррисона. За основу построения матрицы A^{-1} возьмем матрицу B той же размерности, что и A, но с известной обратной матрицей.

Теорема 1. (Вудбери [4]) Для обратимости матрицы A вида $A = B - UV^T$, где A — невырожденная матрица порядка $n \times n$, а U, V пусть $(n \times m)$ — матрицы, необходимо и достаточно, чтобы была обратимой матрица m-го порядка $P = I_m - V^T B^{-1} U$. При этом

$$A^{-1} = B^{-1} + B^{-1}UP^{-1}V^T B^{-1}. \quad (1)$$

В частном случае, при $m = 1$, тождество Вудбери приводит к следующей теореме:

Теорема 2. (Шерман – Моррисон [5]) Пусть B — невырожденная матрица и вектора $\mathbf{u}$ и $\mathbf{v}$ такие, что

$$r = 1 + \mathbf{v}^T B^{-1}\mathbf{u} \neq \mathbf{0}.$$

Тогда матрица $A = B^{-1} + \mathbf{v}^T\mathbf{u}$ является обратимой и её обращение находится как

$$A^{-1} = B^{-1} - r^{-1}B^{-1}\mathbf{u}\mathbf{v}^T B^{-1}. \tag{2}$$

Далее будем рассматривать и считать, что матрица A обратимая с $a_{ii} \neq 0$ и нет необходимости в перестановках при её обращении. В этом случае, при условии $\mathbf{u} = \mathbf{e}_k = [0 \dots 1 \dots 0]^T$, выражение (2) можно упростить, полагая что $B^{-1}\mathbf{e}_k = \mathbf{b}_k^*$, $\mathbf{b}_k^*$ — k-ый столбец матрицы B^{-1}. Тогда столбец $\mathbf{a}_j^*$ матрицы A^{-1} вычисляется

$$\mathbf{a}_j^* = \mathbf{b}_j^* - \frac{\mathbf{v}^T\mathbf{b}_j^*}{1 + \mathbf{v}^T\mathbf{b}_j^*}\mathbf{b}_k^*, \quad j = 1, 2, \dots, N. \tag{3}$$

По соотношениям (3) вычислим обращение матрицы A, принимая что $B = E$ и записывая на k шаге

$$\mathbf{u} = \mathbf{e}_k, \quad \mathbf{v}^T = \mathbf{v}^k = \mathbf{a}^k - \mathbf{e}_k^T, \tag{4}$$

исходя из

$$A = B + \sum_{k=1}^{N} \mathbf{v}^k.$$

Столбец матрицы на k шаге находится из соотношения

$$\mathbf{a}_j^{(k)} = \mathbf{a}_j^{(k-1)} - \frac{\mathbf{v}^T\mathbf{a}_j^{(k-1)}}{1 + \mathbf{v}^T\mathbf{a}_k^{(k-1)}}\mathbf{a}_k^{(k-1)}, \quad j = 1, 2, \dots, N, \tag{5}$$

где $A^{(0)} = E$ и для $k = N$ находится $\mathbf{a}_j^{(N)} = \mathbf{a}_j^*, j = 1, 2, \dots, N$.

Реализация алгоритма (2) в виде (5) приведена в Алгоритме 1. Векторные операции реализовывались на GPGPU с помощью функций библиотеки cuBLAS.

Алгоритм 1 Параллельный Шермана – Моррисона

Require: $\mathbf{a}, \mathbf{v} \in \mathbb{R}^N$ {вектора хранятся в памяти GPU}
1: $\mathbf{v}^T = \mathbf{a}^k - \mathbf{e}_k^T$ { вектор $\mathbf{e}_k \in \mathbb{R}^N$, k элемент которого равен 1, а все остальные 0}
2: dim {размерность матрицы }
3: **for** $k = 0 \to N$ **do**
4: **for** $i = 0 \to N$ **do**
5: **for** $j = 0 \to N$ **do**
6: $\beta_1 \leftarrow (\mathbf{v}_k^T, \mathbf{a}_j^{(k-1)})$ {вычисляется с помощью функции cublasDdot}
7: $\beta_2 \leftarrow 1 + (\mathbf{v}_k^T, \mathbf{a}_k^{(k-1)})$ {cublasDdot}
8: $\beta_3 \leftarrow \beta_1/\beta_2$
9: $\mathbf{a}_j^{(k)} \leftarrow \mathbf{a}_j^{(k-1)} - \beta_3$ {cublasDaxpy}
10: **end for**
11: **end for**
12: **end for**

В работе [7] был предложен блочный алгоритм обращения матриц на основе (1) в случае плотных матриц, в котором требуется достаточно затратное разделение матриц на блоки. Ускорение при использования GPU в этом случае составило лишь 20%.

В настоящей работе рассматривается алгоритм обращения матриц с использованием формул (2) для разряженных симметричных матриц.

2. Численные исследования

Численные эксперименты, представленные в таблице, были проведены на тестовых матрицах известных библиотек [8, 9], а также на матрицах полученных при решении задачи напряженно-деформированного состояния винтовой пружины методом дополнения Шура [3]. Незаполненные поля таблицы соответствуют вычислительным затратам существенно превышающим результаты по другим методам. Размеры матриц варьируются от 276 до 18000, число ненулевых элементов — от 1666 до 6897316, числа обусловленности χ — от 10 до 10^{10}. Все результаты получены с двойной точностью.

Для вычислений использовался узел с четырехядерным процессором Intel Xeon CPU E5430 частотой 2667 МГц и графическим ускорителем NVIDIA GeForce GTX 580. Количество ядер CUDA 512, объем памяти 3 ГБ, частота ядра/памяти 772 МГц/4008 МГц.

Выделим три основных параметра для сравнения и оценки алгоритмов обращения матриц на параллельной вычислительной системе: число арифметических операций, затраты памяти и требования к обработке данных.

Затраты памяти метода сопряженных градиентов и метода пополнения на формирование исходной разреженной и обратной матриц составляют $16NNZ + 4N^2$ и $8N^2$ байт, соответственно. В методе Шермана – Моррисона на вспомогательные переменные, требуется $16N$, в методе сопряженных градиентов затраты несколько больше и составляют $48N$. Доступ к элементам матрицы в формате CSR осуществляется по столбцам в методе SM и по строкам в CG, тогда как в методах Гаусса – Жордана и LU-факторизации доступ к матрице происходит поэлементно и на каждом шаге необходимо иметь всю матрицу. Вычислительные затраты алгоритмов по числу операций выглядят следующим образом: в методе сопряженных градиентов без предобуславливания на каждой k-итерации один раз выполняется матрично-векторное произведение и три скалярных произведения $N(N^2 + 3N + 3)$; в методе пополнения на каждой итерации N раз выполняется два скалярных произведения $N^2(2N + 2)$.

Для выполнения скалярных произведений в методе SM создается вектор в памяти GPU, в который записывается столбец матрицы, необходимый на данном шаге алгоритма. В матрично-векторном произведении для метода CG каждую строку матрицы обрабатывает несколько нитей (до 32) одновременно, что обеспечивает его выполнение в два и, даже, три раза быстрее скалярного произведения при произвольных размерах матриц и их разрежённости.

Определяющей, при использовании метода сопряженных градиентов, становится обусловленность (χ) обращаемой матрицы. В случае с хорошо обусловленными матрицами (когда число итераций метода сопряженных градиентов меньше размера матрицы), число операций в методе сопряженных градиентов будет меньше, чем в методе SM (см. табл. 1).

При плохой обусловленности матриц, но большой разреженности (Dubcova1 16129/253009, bodyy4 17546/121550) CG работает на порядок быстрее, чем метод SM. Од-

Таблица

Время обращения матриц, с

Название	N/Nnz	χ	GJ CPU	*LU* CPU	CG GPU	SM CPU	SM GPU
Schur276	276/7488	1.2e+02	**0.1**	0.1	1.1	0.1	4.1
Lect 494_BUS	494/1666	3.9e+06	**0.3**	0.7	23.9	1.31	12.1
1138_BUS	1138/4054	8.0e+09	**9.5**	11.9	133.1	23.0	62.5
Schur1236	1236/40806	1.5e+01	12.6	18.6	**8.8**	34.5	74.5
BCSSTK11	1473/34241	5.3e+08	**21.4**	33.9	246.6	57.9	104.8
BCSSTK15	3948/117816	8.0e+09	426.0	649.1	**329.1**	1182.9	751.1
Shur5688	5688/203238	2.9e+01	1262.8	1822.5	**84.8**	3171.6	1572.2
ND3K	9000/3279690	1.6e+07	5009.6	7175.4	34801.3	12960.3	**3918.7**
msc10848	10848/1229776	9.9e+09	8718.8	12446.8	15259.6	21663.5	**5741.0**
Dubcova1	16129/253009	9.9e+02	29868.9	40859.7	**617.3**	75472.2	13006.5
bodyy4	17546/121550	8.1e+02	40400.6	—	**376.4**	—	15566.6
ND6K	18000/6897316	1.5e+07	41782.0	35674.5	—	—	**16386.0**

нако, с увеличением числа ненулевых элементов затраты возрастают (ND3K 9000/3279690, ND6K 18000/6897316), а девятикратное ускорение достигается в параллельном методе SM.

В численных экспериментах, для выполнения базовых операций рассматриваемых методов, вызывались функции библиотеки cuBLAS (см. Алгоритм 1). Особенностью применения функции `cublasDdot` является то, что при использовании текстурной памяти в основной программе последующий вызов функции `cublasDdot` выполняется существеннее медленнее, чем другие вызовы при вычислении скалярных произведений. С увеличением размерности векторов это влияние становится значительным и операция скалярного произведения выполняется, практически, в семь раз медленнее, что характерно для метода CG, при реализации которого текстурная память задействована в матрично-векторном произведении.

Используемая реализация алгоритма CG хорошо масштабируется на GPU и позволяет достигать ускорения в десятки раз, по сравнению с вариантом, реализованном на CPU [1], а параллельная реализация алгоритма SM на GPU обеспечивает гораздо меньшие ускорения. Так, для матриц малых размерностей (Schur276, 494_BUS, 1138_BUS, Schur1236, BCSSTK11) время выполнения варианта, реализованного на GPU, больше, чем на CPU, а на матрицах больших размерностей получено максимальное шестикратное ускорение. Прежде всего это связано с лучшей масштабируемостью скалярного произведения небольших векторов на CPU, а не на GPU. В случае метода сопряженных градиентов значительное ускорение вычислений на GPU обеспечивает матрично-векторное произведение, которое обладает большой степенью параллелизма.

Заключение

При использовании GPU на малых задачах (до $N = 3000$) сокращение времени выполнения операций не покрывают затраты на выделение памяти, поэтому эффективность параллельных реализаций на CPU в два раза выше. Однако, с увеличением размеров матриц, алгоритмы реализованные на CPU не показывают значимых ускорений, а результаты

полученные на GPU, методами сопряженных градиентов и пополнения, эффективнее более чем в десять раз (bodyy4).

Дальнейшие развитие высопроизводительных методов обращения матриц связано с модификацией и оптимизацией алгоритма метода Шермана – Моррисона при обращении симметричных матриц, в том числе и для построения эффективных предобуславливателей на его основе для метода сопряженных градиентов.

Работа выполнена при финансовой поддержке РФФИ (грант 14-01-31066-мол_а, 14-01-00055-а, 13-01-00101-а) и программы Президиума РАН №18 при поддержке УрО РАН (проект 12-П-1-1005).

Литература

1. Копысов, С.П., Кузьмин, И.М., Недожогин, Н.С., Новиков, А.К. Параллельные алгоритмы формирования и решения системы дополнения Шура на графических ускорителях // Ученые записки Казанского университета. Серия Физ.-мат. науки – 2012. – Т. 154. №3. – С. 202-215.
2. Ezzatti, P., Quintana-Orti, E.S., Remon, A. Using graphics processors to accelerate the computation of the matrix inverse // J. of Supercomputing. – 2011. V.58. – P.429-437.
3. Kopysov, S.P., Kuzmin, I.M., Nedozhogin, N. S., Novikov, A.K., Sagdeeva, Y.A. Hybrid Multi-GPU solver based on Schur complement method // Lecture Notes in Computer Science – 2013. – vol. 7979. – pp. 65-79.
4. Woodbury, M. Inverting modified matrices // Memorandum Rept. 42, Statistical Research Group, Princeton University, Princeton, NJ, 1950.
5. Sherman, J., Morrison, W.J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix // Ann. Math. Statistics. – 1950. – V. 21. №1. – P. 124-127.
6. Фаддеев, Д.К., Фаддеева, В.Н. Вычислительные методы линейной алгебры. – СПб.: Изд-во "Лань 2002. – 736 с.
7. He, X., Holm, M., Neytcheva, M. Efficiently parallel implementation of the inverse Sherman – Morrison algorithm // Lecture Notes in Computer Science. – 2013. – V. 7782. — P. 206-219.
8. Matrix Market: URL: http://math.nist.gov/MatrixMarket/ (дата обращения: 29.08.2013)
9. Davis, T. University of Florida Sparse Matrix Collection: sparse matrices from a wide range of applications: URL: http://www.cise.ufl.edu/research/sparse/matrices/ (дата обращения: 29.08.2013)

Недожогин Никита Сергеевич, аспирант, Институт механики УрО РАН (Ижевск, Российская Федерация), Nedozhogin@inbox.ru.

Сармакеева Анастасия Семеновна, студент математического факультета, Удмуртский государственный университет (Ижевск, Российская Федерация), asarmakeeva@gmail.com.

Копсов Сергей Петрович, д.ф.-м.н., профессор, заведующий лабораторией, Институт механики УрО РАН (Ижевск, Российская Федерация), s.kopysov@gmail.com.

Поступила в редакцию 14 марта 2014 г.

Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 2, pp. 101–108

SHERMAN – MORRISON HIGH-PERFORMANCE ALGORITHM FOR INVERSE MATRIX ON GPU

N.S. Nedozogin, Institute of Mechanics of Ural Branch of the RAS (Izhevsk, Russian Federation),
A.S. Sarmakeeva, Institute of Mechanics of Ural Branch of the RAS (Izhevsk, Russian Federation),
S.P. Kopysov, Institute of Mechanics of Ural Branch of the RAS (Izhevsk, Russian Federation)

Matrix inversion is widely used in numerical methods, such as linear solvers, preconditioning for linear system, domain decomposition, digital image processing, etc. High-performance implementation of matrix inversion requires efficient matrix storage formats and optimal distribution of computations between computing devices. In this paper, we study the performance of traditional matrix inversion algorithms, such as LU-factorization and Gauss-Jordan, as well as the conjugate gradient method and the Sherman – Morrison formula. In the last two algorithms, matrix-vector products and scalar products are efficiently executed on multicore/manycore processors. We compare the performance of the algorithms on hybrid multi-CPU multi-GPU platforms, using the matrices from well-know test suites and from the numerical simulation of wrap spring.

Keywords: Sherman – Morrison formula, high-performance computing, inverse matrix, sparse matrix.

References

1. Kopysov S.P., Kuzmin I.M., Nedozhogin N.S., Novikov A.K. Parallelnye algoritmy formirovaniya i resheniya sistemy dopolneniya Shura na graficheskix uskoritelyax // Uchenye zapiski Kazanskogo universiteta. Seriya Fiz.-mat. nauki – 2012. – T. 154. №3. – P. 202-215.
2. Ezzatti P., Quintana-Orti E.S., Remon A. Using graphics processors to accelerate the computation of the matrix inverse // J. of Supercomputing. – 2011. V.58. – P.429–437.
3. Kopysov S.P., Kuzmin I.M., Nedozhogin N.S., Novikov A.K., Sagdeeva Y.A. Hybrid Multi-GPU solver based on Schur complement method // Lecture Notes in Computer Science – 2013. – vol. 7979. – pp. 65–79.
4. Woodbury, M. Inverting modified matrices // Memorandum Rept. 42, Statistical Research Group, Princeton University, Princeton, NJ, 1950.
5. Sherman J., Morrison W.J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix // Ann. Math. Statistics. – 1950. – V. 21. №1. – P. 124–127.
6. Faddeev D.K., Faddeeva V.N. Vychislitelnye metody linejnoj algebry. – SPb.: Lan, 2002. – 736 p.
7. He X., Holm M., Neytcheva M. Efficiently parallel implementation of the inverse Sherman – Morrison algorithm //Lecture Notes in Computer Science. – 2013. – V. 7782. — P. 206–219.

8. Matrix Market: URL: http://math.nist.gov/MatrixMarket/ (accessed: 29.08.2013)

9. Davis, T. University of Florida Sparse Matrix Collection: sparse matrices from a wide range of applications: URL: http://www.cise.ufl.edu/research/sparse/matrices/ (accessed: 29.08.2013)

Received 14 March 2014.

УДК 519.687

СОЗДАНИЕ СИМУЛЯТОРА ДЛЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ОБРАЗОВАТЕЛЬНОГО РЕСУРСНОГО ЦЕНТРА [1]

П.Н. Полежаев, А.Л. Коннов, А.Е. Шухман

Данная статья посвящена проблеме создания образовательных ресурсных центров, как средства организации экономически выгодного удаленного доступа образовательных учреждений к платному программному обеспечению. ЦОД образовательного ресурсного центра построен на базе инновационной технологии программно-конфигурируемых сетей, что позволяет эффективно решать проблемы маршрутизации и обеспечения качества обслуживания. Для исследования алгоритмов планирования виртуальных классов и виртуальных машин образовательного ресурсного центра необходимо создание его симулятора. С помощью диаграмм UML описывается проект создания симулятора – диаграмма вариантов использования показывает его функциональные возможности, а диаграмма деятельности – принцип его функционирования. Также разработана имитационная модель образовательного ресурсного центра, включая имитационные схемы всей системы в целом, вычислительного узла, коммутатора OpenFlow и контроллера OpenFlow.

Ключевые слова: облачные вычисления, DaaS, образовательные ресурсные центры, имитационное моделирование, планирование виртуальных машин, программно-конфигурируемые сети, OpenFlow.

Введение

В настоящее время в развивающихся странах большинство образовательных учреждений функционирует в условиях недостаточного финансирования на закупку ПО, необходимого в учебном процессе. Это приводит к снижению качества образования, а также к росту уровня незаконного использования ПО.

В рамках данной работы предлагается решение этой проблемы путем создания образовательных ресурсных центров (ОРЦ) с возможностью удаленного доступа к ним образовательных учреждений. Каждый ОРЦ представляет собой центр обработки данных, поверх него развернута облачная система OpenStack [1] с виртуальными машинами, содержащими все необходимое платное ПО (см. рис. 1).

Каждое образовательное учреждение, зарегистрированное в ОРЦ, получает доступ к одному или нескольким виртуальным компьютерным классам, представляющим собой группы экземпляров виртуальных машин со всем необходимым установленным платным и бесплатным программным обеспечением. Доступ предоставляется учащимся по сети Интернет, при этом сами учащиеся используют компьютеры физических классов образовательных учреждений или собственные индивидуальные устройства – ноутбуки, смартфоны, планшеты.

Использование подобных виртуальных классов позволяет образовательным учреждениям не тратиться на закупку дорогостоящего ПО, необходимого в учебном процессе. ОРЦ предоставляет его в аренду в рамках виртуальных классов на время проведения

[1] Рекомендована к публикации программным комитетом научной конференции «Параллельные вычислительные технологии 2014»

занятий, при этом ПО может использоваться совместно в нескольких виртуальных классах с учетом лицензионных ограничений. Кроме того, образовательным учреждениям нет необходимости владеть мощными современными компьютерами. Они должны только поддерживать работу браузера для удаленного доступа к виртуальным классам и иметь быстрое подключение к сети Интернет, что приводит к значительной экономии денежных средств.

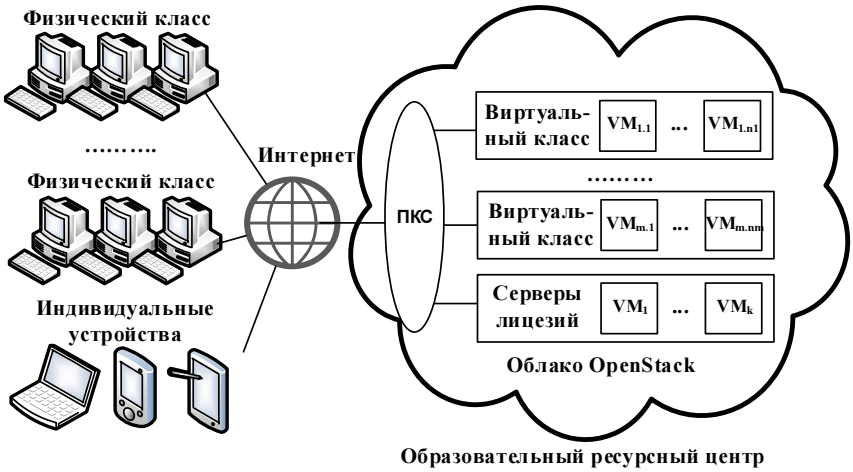


Рис. 1. Логическая структура образовательного ресурсного центра

Помимо виртуальных классов в облаке OpenStack развернуто несколько виртуальных машин, содержащих сервера лицензий для платного ПО с лицензиями, ограничивающими количество одновременно работающих в сети экземпляров программ.

1. Проектирование симулятора образовательного ресурсного центра

Основные функции симулятора ОРЦ отражены на диаграмме вариантов использования UML, приведенной на рис. 2.

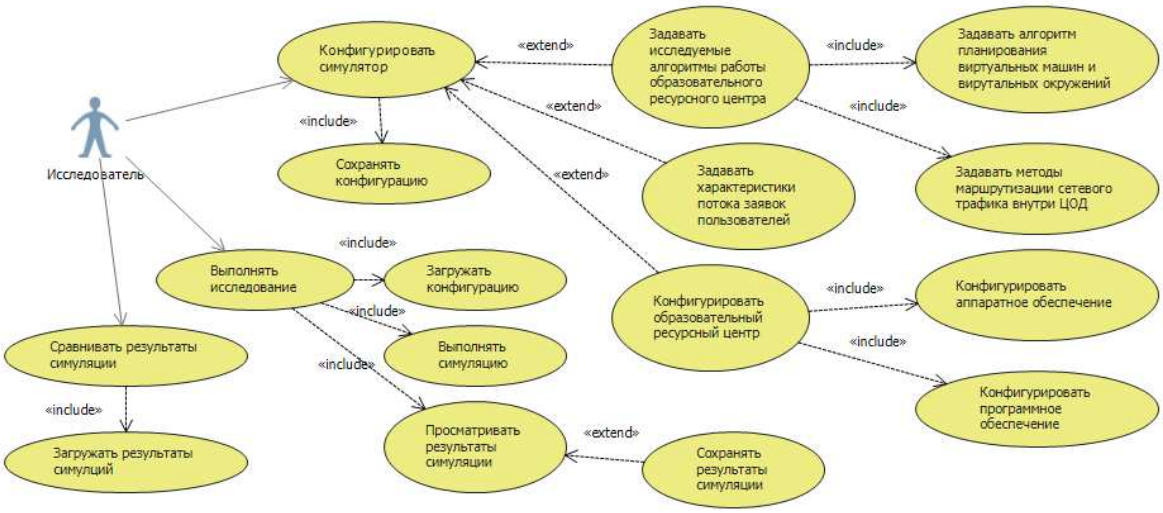


Рис. 2. Диаграмма вариантов использования для симулятора ОРЦ

В их число входит:

а) Конфигурирование симулятора, которое позволяет: задавать исследуемые алгоритмы работы ОРЦ (включая алгоритмы планирования виртуальных машин и виртуальных окружений, методы маршрутизации сетевого трафика внутри ЦОД); задавать характеристики потока заявок пользователей; конфигурировать аппаратное и программное обеспечение симулируемого образовательного ресурсного центра. Построенная конфигурация может быть сохранена в файл и многократно использована в проводимых исследованиях.

б) Выполнение исследования, загрузка из файла созданной конфигурации, выполнение симуляции, отображение результатов симуляции с возможностью их сохранения в файл.

в) Сравнение результатов симуляции, путем их загрузки из файлов с результатами и сопоставления между собой.

На рис. 3 изображена диаграмма деятельности, описывающая обобщенный алгоритм работы симулятора образовательного ресурсного центра

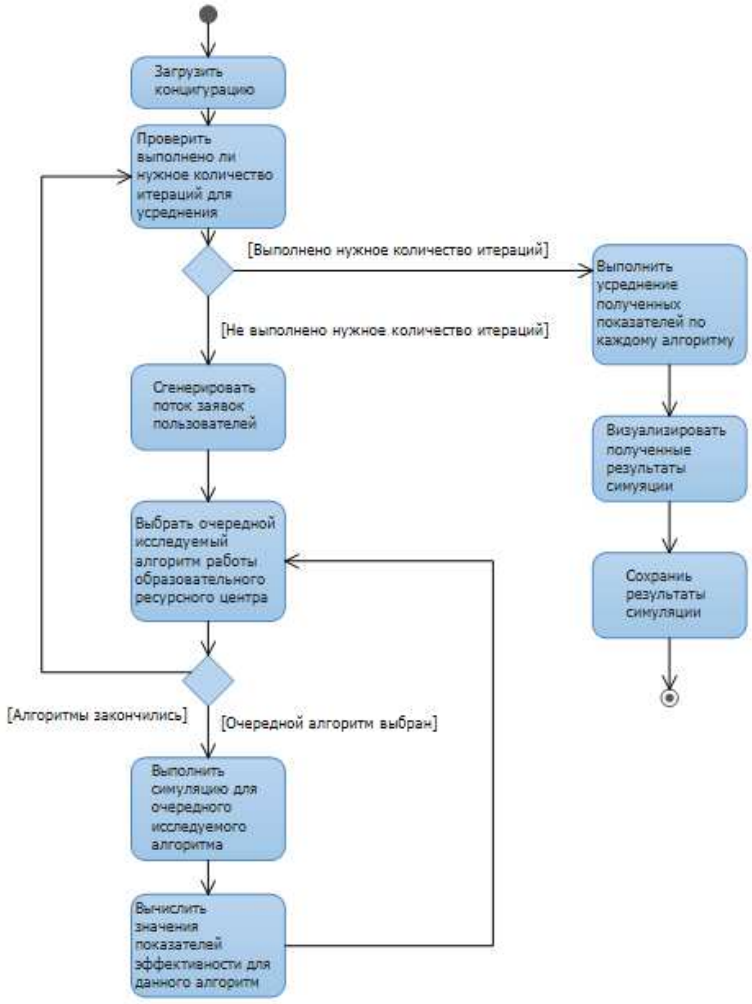


Рис. 3. Диаграмма деятельности, описывающая алгоритм работы симулятора ОРЦ

В начале происходит загрузка созданной конфигурации, затем выполняется цикл, на каждой его итерации генерируется поток заявок пользователей, на котором исследуется каждый из указанных в конфигурации алгоритмов работы образовательного ресурсного

центра. Для каждого алгоритма и потока заявок пользователей собираются значения показателей производительности, затем они усредняются по всем сгенерированным потокам задач. Полученные результаты визуализируются и сохраняются в файл с результатами.

2. Разработка симулятора образовательного ресурсного центра

Для экспериментального исследования эффективности разрабатываемых алгоритмов планирования виртуальных машин и виртуальных окружений, а также методов маршрутизации данных, необходимо создание симулятора образовательного ресурсного центра, основанного на ПКС. В его основе лежит имитационная модель, описываемая в данном разделе. Она включает:

а) моделирование обработки заявок пользователей;

б) моделирование сетевого трафика в ПКС, возникающего в процессе обработки заявок.

На рис. 4 приведена имитационная схема процесса обработки заявок пользователей.

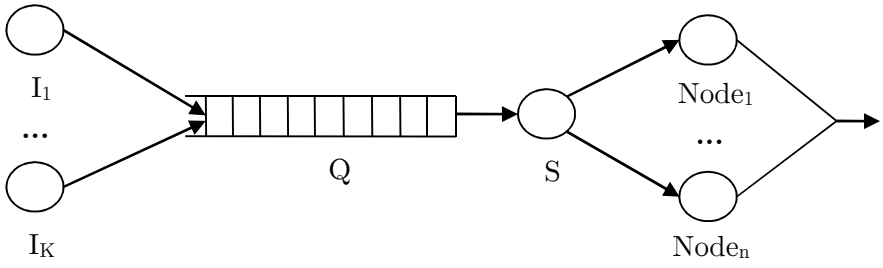


Рис. 4. Имитационная схема процесса обработки заявок пользователей

Каждый из источников заявок $I_1,...,I_K$ представляет собой образовательное учреждение, подключенное к ресурсному центру. $I_1,...,I_K$ совместно генерируют поток пользовательских заявок на обслуживание $J = \{J_j\}_{j=\overline{1,R}}$. Каждая заявка J_j имеет следующий вид:

$$J_j = (a_j, t_j, Classroom_j), \quad (1)$$

где a_j – время поступления заявки, t_j – продолжительность времени, в течение которого виртуальный класс $Classroom_j$ должен быть доступен образовательному учреждению после развертывания. По истечению времени t_j виртуальный класс должен выгрузиться с физических узлов ЦОД образовательного ресурсного центра.

Все поступающие от пользователей заявки помещаются в очередь Q. Канал S представляет собой планировщик, который в соответствии с заложенным в него алгоритмом осуществляет извлечение заявок J_j из очереди Q, резервирование для них места в расписании (осуществляется выбор подмножества узлов $A_j \subseteq Nodes$ и момента времени T_j для их запуска). Набор всех резервирований формирует расписание образовательного ресурсного центра:

$$Schedule = \{(J_j, A_j, T_j)\}_{j=\overline{1,R}}. \quad (2)$$

Обработка заявки J_j заключается в запуске соответствующего виртуального класса и при необходимости дополнительных виртуальных машин с серверами лицензирования

на узлах множества A_j в текущий момент времени t и в его выгрузке в момент времени $t+T_j$.

Имитационная схема вычислительного узла p_i приведена на рис. 5, p_i соединен с другими узлами и коммутаторами вычислительной сети с помощью r_i дуплексных связей. Все входящие пакеты, а также пакеты сообщений, генерируемых назначенными узлу виртуальными машинами $\chi_{i,j}$, сначала поступают в очередь $Q_{inp.i}$, а затем маршрутизируются каналом обслуживания R_i. Если соответствующий пакет предназначен для локальному экземпляру виртуальной машины, то он передается ему непосредственно, иначе – помещается в одну из очередей $Q_{out.i,1}, Q_{out.i,2}, \ldots, Q_{out.i,r_i}$, соответствующую выбранной алгоритмом маршрутизации исходящей связи.

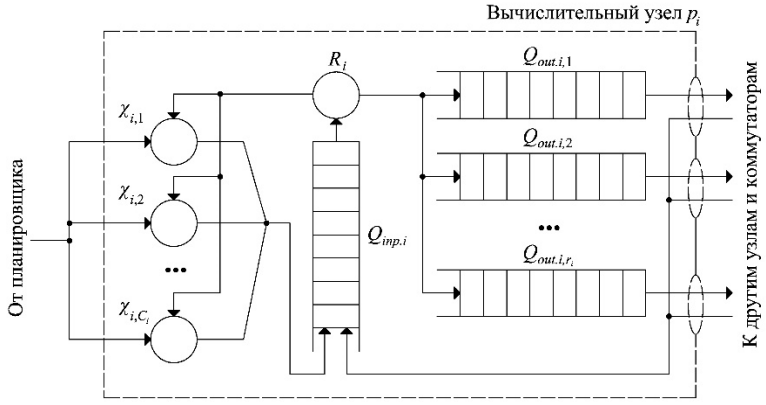


Рис. 5. Имитационная схема вычислительного узла

При выполнении маршрутизации каналом R_i моделируется временная задержка. Заметим, если две виртуальные машины выполняются на одном узле, то обмен пакетами между ними также осуществляется через очередь $Q_{inp.i}$ и канал R_i.

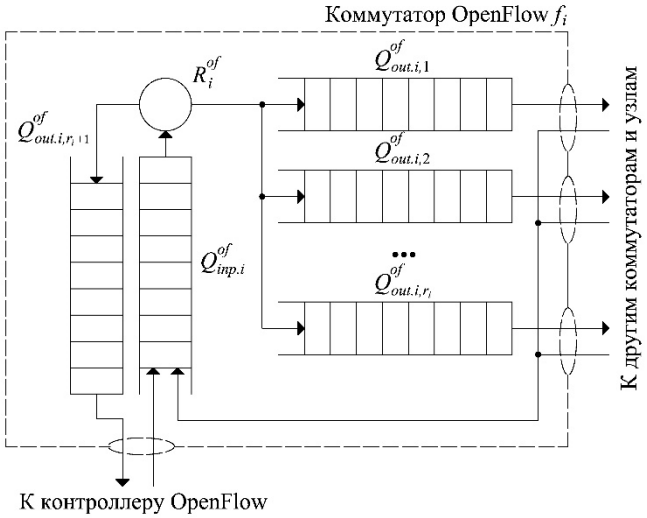


Рис. 6. Имитационная схема коммутатора OpenFlow

Коммутатор OpenFlow f_i (см. **рис. 6**) отличается от обычного коммутатора тем, что его канал обслуживания R_i^{of} реализует поведение модуля OpenFlow – при получении нового пакета поиск и выполнение правил в таблице потоков, в случае отсутствия подходящего правила R_i^{of} отправляет пакет контроллеру OpenFlow, поместив его в очередь $Q_{out.i,r_i+1}^{of}$. Ответные команды контроллера в виде пакетов помещаются в очередь $Q_{inp.i}^{of}$, в которой для них задается максимальный приоритет обслуживания.

Контроллер OpenFlow установлен на выделенный сервер, его имитационная схема аналогична имитационной схеме вычислительного узла (см. рис. 7).

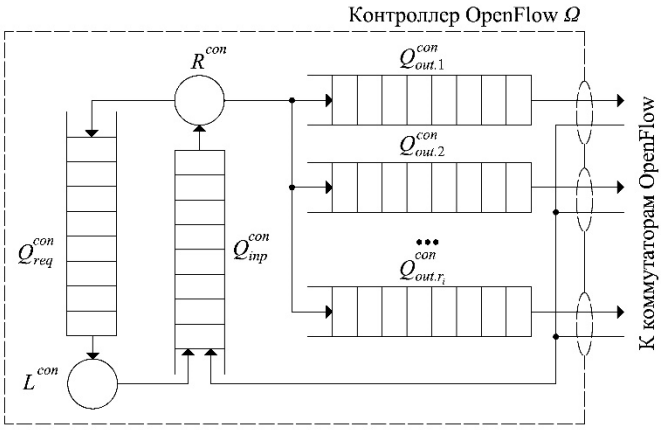


Рис. 7. Имитационная схема контроллера OpenFlow

Отличие в том, что канал R^{con}, помимо стандартной маршрутизации пакетов, помещает запросы к контроллеру OpenFlow в отдельную очередь Q_{req}^{con}. Канал обслуживания L^{con} реализует логику работы контроллера по принятию решений о дальнейших действиях с поступившим пакетом. Результатом его работы являются управляющие пакеты OpenFlow, передаваемые через Q_{inp}^{con} и R^{con} коммутаторам OpenFlow с целью отправки пакета на определенный порт или установки новых правил в их таблицы потоков. В Q_{inp}^{con} и очередях портов такие пакеты имеют максимальный приоритет.

Поток заявок на обслуживания определяется случайными величинами с законами распределения и параметрами, выбранными на основе существующих исследований [2]. Реализация данной модели в виде симулятора высокопроизводительной системы позволит исследовать эффективность предлагаемых в рамках проекта алгоритмических решений.

Заключение

В рамках настоящего исследования предложена концепция ОРЦ, как средства организации экономически выгодного удаленного доступа образовательных учреждений к платному ПО. С помощью диаграмм UML описаны функциональные возможности и принцип работы симулятора ОРЦ, который будет использован для исследования алгоритмов планирования виртуальных классов и виртуальных машин. Также разработана имитационная схема функционирования ОРЦ в целом и его основных узлов.

Исследования выполнены при поддержке РФФИ (проекты №13-07-97046 и №14-07-97034).

Литература

1. OpenStack Open Source Cloud Computing Software // Rackspace Cloud Computing, 2013. URL: http://www.openstack.org/ (дата обращения: 21.03.2014)
2. Feitelson, D. Workload Modeling for Computer Systems Performance Evaluation // Удаленный ресурс Еврейского университета Иерусалима. URL: http://www.cs.huji.ac.il/~feit/wlmod/ (дата обращения: 21.03.2014)

Полежаев Петр Николаевич, преподаватель кафедры компьютерной безопасности и математического обеспечения информационных систем, Оренбургский государственный университет (Оренбург, Российская Федерация), newblackpit@mail.ru

Коннов Андрей Леонидович, кандидат технических наук, доцент кафедры системного анализа и управления, Оренбургский государственный университет (Оренбург, Российская Федерация), andrey_konnov@mail.ru

Шухман Александр Евгеньевич, кандидат педагогических наук, заведующий кафедры геометрии и компьютерных наук, Оренбургский государственный университет (Оренбург, Российская Федерация), shukhman@gmail.com

Поступила в редакцию 10 апреля 2014 г.

Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 2, pp. 109–116

DEVELOPING SIMULATOR OF EDUCATIONAL RESOURCE DATACENTER

P.N. Polezhaev, Orenburg State University (Orenburg, Russian Federation),

A.L. Konnov, Orenburg State University (Orenburg, Russian Federation),

A.E. Shukhman, Orenburg State University (Orenburg, Russian Federation)

This paper is devoted to the development problem of educational resource datacenter. It can be used by educational organizations for cheap remote access to the paid software. Its datacenter is based on innovative technology of software defined networks, which efficiently solve the problems of traffic routing and QoS. Simulator of educational resource datacenter is developed for experimental research of virtual classroom and virtual machines scheduling algorithms. Project of simulator is described by UML diagrams – use case diagram for its functionality, activity diagram for the principle of its operation. Simulation model of educational resource datacenter is created for simulator. It includes simulation schemes of the whole system, computing node, OpenFlow switch and OpenFlow controller.

Keywords: cloud computing, DaaS, educational resource datacenters, simulation, scheduling of virtual machines, software defined networks, OpenFlow.

References

1. OpenStack Open Source Cloud Computing Software // Rackspace Cloud Computing, 2013. URL: http://www.openstack.org/ (accessed: 21.03.2014)
2. Feitelson, D. Workload Modeling for Computer Systems Performance Evaluation // The Hebrew University of Jerusalem. URL: http://www.cs.huji.ac.il/~feit/wlmod/ (accessed: 21.03.2014)

Received 10 April 2014.

УДК 681.3

АВТОМАТИЧЕСКОЕ ОТОБРАЖЕНИЕ ПРОГРАММ НА ПРОЦЕССОР С ПЛИС-УСКОРИТЕЛЕМ

Д.В. Дубров, А.С. Рошаль, Б.Я. Штейнберг, Р.Б. Штейнберг

В работе рассматривается задача автоматического отображения высокоуровневых программ на процессор с ПЛИС-ускорителем. Для такого отображения разрабатывается и используется генератор HDL-кода из внутреннего представления распараллеливающей системы.

Ключевые слова: конвейерные вычисления, высокоуровневый синтез, распараллеливающий компилятор, ПЛИС, VHDL.

Введение

Конвейерные вычислители используются во многих программно-аппаратных комплексах и показывают на некоторых задачах производительность значительно более высокую, чем универсальные процессоры. Существенный прогресс в развитии конвейерных вычислителей внесли технологии программируемых логических интегральных схем (ПЛИС). Традиционно, конвейерные вычислители используются в программно-аппаратных комплексах. Для применения к широкому классу программ разрабатываются вычислители с архитектурой программируемого конвейера или реконфигурируемые конвейеры. Известны работы по реконфигурируемым конвейерам [1, 3, 4]. Узкое место таких вычислителей – долгое время перепрограммирования ПЛИС. В данной работе описывается проект, разрабатываемый на мехмате ЮФУ на основе распараллеливающей системы ДВОР (Диалоговый высокоуровневый оптимизирующий распараллеливатель программ). Этот проект ориентирован на расширение множества разработчиков и пользователей прикладных программ для параллельно-конвейерных суперкомпьютеров. Основная идея проекта связана с созданием генератора VHDL-кода из внутреннего представления распараллеливающей системы.

1. Конвертор программ языка Си в язык VHDL

На входе конвертора текст на языке Си, описывающий алгоритм вычислений создаваемого генератора. Вводятся также некоторые характеристики создаваемой схемы: диапазоны значений входных и выходных переменных, допустимая площадь на кристалле. На выходе HDL-описание электронной схемы. Генератор позволяет создавать встроенные системы, конвейерные вычислители, параллельно-конвейерные вычислители, автоматы, системы с программируемой архитектурой и др. По диапазонам значений входных и выходных данных, определяются диапазоны промежуточных переменных и определяется минимальная разрядность аргументов вычислительных устройств проектируемой схемы. Генератор имеет свою библиотеку VHDL-описаний базовых операций и функций. В этой библиотеке представлены базовые операции и функции для аргументов различной разрядности, которые сопровождаются оценками количества тактов и занимаемой площади кристалла [5].

В настоящее время на рынке представлен целый ряд программных продуктов, обладающих возможностью преобразования высокоуровневых языков в языки описания электронных схем. В следующей таблице приведена информация об основных из них. Как можно видеть, к данному направлению проявляют интерес как небольшие исследовательские

коллективы (разработки C-to-Verilog, CoolKit), так и ведущие разработчики средств проектирования электронных устройств (Xilinx, Mentor Graphics и др.) В последнем случае наблюдается общая для ИТ-индустрии тенденция, выраженная в приобретении крупными компаниями интеллектуальной собственности независимых разработчиков. Так например, язык высокого уровня Handel-C был изначально разработан в вычислительной лаборатории Оксфордского университета [6], после чего в 1996 г. был коммерциализирован созданной компанией Embedded Solution Limited (переименованной позднее в Celoxica), которая в 2008 г. была приобретена компанией Agility, чьи активы год спустя были в свою очередь приобретены Mentor Graphics. В настоящее время Handel-C поддерживается их продуктом DK Design Suite. Другим подобным примером является язык Streams-C, разработанный в Лос-Аламосской национальной лаборатории и лицензированный компании Impulse Accelerated Technologies под именем Impulse C.

2. Диалоговый высокоуровневый оптимизирующий распараллеливатель программ (ДВОР)

Одна из особенностей распараллеливающей системы ДВОР – высокоуровневое внутреннее представление, именуемое Reprise. Генерация описания конвейерных электронных схем из регистровых внутренних представлений (GCC, LLVM) представляется очень сложной. Система ДВОР содержит автоматическое построение графа вычислений программы [4], который является промежуточным представлением между Reprise и HDL-описанием конвейерной схемы.

Привязка конвертора языка программирования Си в язык описания электронных схем к распараллеливающей системе дает возможность использования библиотеки преобразований высокоуровневых программ. Благодаря этому, исходный код программы Си может быть преобразован в другие эквивалентные программы. Это позволяет по исходному коду сгенерировать несколько эквивалентных электронных схем, а не одну единственную. Это важно, поскольку к электронным схемам предъявляются различные требования, иногда противоречащие одно другому: минимизация площади кристалла, быстродействие, энергопотребление,...

3. Отображение гнезд циклов на программируемый вычислитель

Параллельные вычисления, в том числе и конвейерные, следует применять к долго считаемым фрагментам программ. В данной работе будем рассматривать ускорение фрагментов кода, содержащих гнезда программных циклов. Конвейеризуется самый глубоко вложенный цикл гнезда. Счетчики более высоко вложенных циклов могут рассматриваться как параметры, определяющие узел кластера, на ускорителе которого должен выполняться самый глубоко вложенный цикл (этот же цикл для разных значений счетчиков внешних циклов выполняется на разных узлах).

Основная идея проекта следующая. Разрабатываемая система получает на входе программу языка Си, содержащую гнездо циклов. Гнездо циклов преобразуется к виду, удобному для отображения распараллеливающей системой на конвейерную архитектуру. Самый вложенный цикл гнезда преобразуется в VHDL-описание конвейера конвертором C2HDL.

Таблица

Обзор основных средств, преобразующих высокоуровневые программы в HDL

Название	Разработчик	Время разработки	Тип лицензии	Входные языки	Выходные языки
Catapult C	Mentor Graphics (http://www.mentor.com/esl/catapult/overview)	с 2004 г.	коммерческая	C, C++	RTL
Vivado Design Suite	Xilinx (http://www.xilinx.com/products/design-tools/vivado/)	с 2012 г.	коммерческая	C, C++, MATLAB, Simulink	VHDL, Verilog
Impulse CoDeveloper	Impulse Accelerated Technologies (http://www.impulseaccelerated.com/products.htm)	с 2003 г.	коммерческая	C, Impulse C	VHDL, Verilog
Altium Designer	Altium (http://www.altium.com/en/products/altium-designer)	с 2008 г.	коммерческая	C	VHDL, Verilog
Mitrion SDK	Mitrionics (http://www.mitrionics.com/?page=mitrion-sdk)	с 2005 г.	коммерческая	Mitrion-C	VHDL
MATLAB + Simulink + HDL Coder	MathWorks (http://www.mathworks.com/products/simulink/)	с 2007 г.	коммерческая	MATLAB, Simulink, Stateflow	VHDL, Verilog
C-to-Verilog	Nadav Rotem (Haifa University) (http://www.c-to-verilog.com/)	с 2009 г.	GPL	C	Verilog
CoolKit	СПбГУ (http://oops.math.spbu.ru/projects/coolkit)	с 2008 г.	“No license”	HaSCoL	VHDL

Программируемая часть вычислительного узла прожигается полученной схемой. В исходном гнезде циклов самый вложенный цикл заменяется обращением к конвейерному ускорителю, на котором этот цикл будет вычисляться.

Условие эффективности использования ПЛИС-ускорителя – время передачи данных на ускоритель должно быть существенно меньше времени обработки этих данных. Учитывая, что при современных технологиях время пересылки одного четырехбайтного числа между чипами на порядок дольше умножения чисел, можно заключить, что в ускоряемом фрагменте кода количество операций над данными должно в несколько десятков раз пре-

восходить количество данных. Значит, конвейеризуемый самый вложенный цикл должен содержать в своем теле несколько десятков операций. Необходимое для эффективности отношение количества операций к количеству данных существенно снижается при использовании коротких чисел (например, двух- или однобайтных).

Заключение

В данной работе частично реализованный проект создания компилятора с языка высокого уровня на процессор с ПЛИС-ускорителем. Этот проект ведет к решению проблемы создания высокоуровневых языков для вычислительных устройств с архитектурой программируемого конвейера.

Литература

1. Каляев, А.В. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений / А.В. Каляев, И.И. Левин — М., «Янус-К», 2003. — 380 с.
2. Штейнберг, Р.Б. Отображение гнезд циклов на многоконвейерную архитектуру / Р.Б. Штейнберг. // Программирование. — 2010. — № 3.
3. Яджак, М.С. Высокопараллельные алгоритмы и методы для решения задач массовых арифметических и логических вычислений / М.С. Яджак. Диссертация на соискание ученой степени д.ф.-м.н. — Институт прикладных проблем механики и математики, Львов, 2009. — 298 с. (на украинском языке).
4. Bondalapati, K. Modeling and Mapping for Dynamically Reconfigurable Hybrid Architecture. / K. Bondalapati. Ph.D. Thesis — University of Southern California, August 2001.
5. Dubrov, D.V. Generating Pipeline Integrated Circuits Using C2HDL Converter D.V. Dubrov, A.S. Roshal // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2013), Rostov-on-Don, Russia, September 27-30, 2013 — 2013. — P. 215–219.
6. Self, R.P. A Design Methodology for Construction of Asynchronous Pipelines with Handel-C / R.P. Self, M. Fleury, A.C. Downton. — IEEE Software, 1988. — Vol. 150, P. 39–47.

Дубров Денис Владимирович, к.ф.-м.н., доцент, Южный федеральный университет (г. Ростов-на-Дону, Российская Федерация), dubrov@sfedu.ru.

Рошаль Александр Сергеевич, ведущий программист, tabor.ru (г. Ростов-на-Дону, Российская Федерация), teacplusplus@gmail.com.

Штейнберг Борис Яковлевич, д.т.н., доцент, зав. каф., Южный федеральный университет (г. Ростов-на-Дону, Российская Федерация), borsteinb@mail.ru.

Штейнберг Роман Борисович, к.ф.-м.н., ст. преподаватель Южный федеральный университет (г. Ростов-на-Дону, Российская Федерация), romanofficial@yandex.ru.

Поступила в редакцию 7 марта 2014 г.

Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 2, pp. 117–121

AUTOMATIC MAPPING PROGRAMS ONTO A PROCESSOR WITH AN FPGA ACCELERATOR

D.V. Dubrov, Southern Federal University (Rostov-on-Don, Russian Federation),
A.S. Roshal, Southern Federal University (Rostov-on-Don, Russian Federation),
B.Ya. Steinberg, Southern Federal University (Rostov-on-Don, Russian Federation),
R.B. Steinberg, Southern Federal University (Rostov-on-Don, Russian Federation)

A problem of automatic high level program mapping onto a CPU with an FPGA accelerator is considered in this work. For such a mapping an HDL code generator from a parallelizing system's internal representation is being developed and used.

Keywords: social network analysis, information retrieval, data mining, expert finding, popularity analysispeline computing, high-level synthesis, parallelizing compiler, FPGA, VHDL.

References

1. Kalyaev A.V., Levin I.I. Modul'no-naraschivaemye mnogoprocessornye sistemy so structurno-procedurnoj organizatsyej vychislenij [Modular scalable multiprocessor systems with structural-procedural computing setup]. Moscow, Yanus-K, 2003. 380 P.
2. Steinberg R. Mapping loop nests to multipipelined architecture. MAIK Nauka/Interperiodica distributed exclusively by Springer Science+Business Media LLC., May 2010, Vol. 36, No. 3, P 177–185.
3. Yadzhak M.S. Vysokoparallel'nye algoritmy i metody dlya resheniya zadach massovykh arifmeticheskikh i logicheskikh vychislenij [Highly parallel algorithms and methods for solving problems with massive arithmetic and logic computations]. Dissertation for a degree of doctor of physics and mathematics, Institut prikladnykh problem mekhanili i matematiki [Institute of applied mechanics and mathematics problems]. Lvov, 2009. 298 P. (in Ukrainian language).
4. Bondalapati K. Modeling and Mapping for Dynamically Reconfigurable Hybrid Architecture. Ph.D. Thesis. University of Southern California, August 2001.
5. Dubrov D.V., Roshal A.S. Generating Pipeline Integrated Circuits Using C2HDL Converter Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2013), Rostov-on-Don, Russia, September 27-30, 2013, P. 215–219.
6. Self R.P., Fleury M., Downton A.C. A Design Methodology for Construction of Asynchronous Pipelines with Handel-C. IEEE Software, 1988, Vol. 150, P. 39–47.

Received 7 March 2014.

IMPACT OF WRF-3DVAR DATA ASSIMILATION ON THE PREDICTION OF RAINFALL OVER SOUTHERN BRAZIL

V.C. Beck, Y. Yamasaki, F.P. Härter

The procedure to combine mathematical models with noise data, in order to improve numerical weather forecasting by statistical methods, is an important and challenging meteorology research field, known as data assimilation. The 3DVAR approach, state of the art in data assimilation technique, is applied in this study. The aim of present development is to evaluate the results of the data assimilation from INMET automatic weather stations and atmospheric soundings in the weather forecast produced by WRF model. The region of interest is the South of Brazil. The specific aim is to evaluate the assimilation procedure of two precipitation events occurred in 2012. This study is especially important, because the INMET automatic weather stations data are not transmitted by GTS. Therefore, these data were not assimilated by prediction systems generated by global models, such as GFS, which provides initial and boundary conditions for regional models, such as WRF. The results showed that the WRF with data assimilation procedure, reproduces satisfactorily the true synoptic scenario given by GFS model in the two cases evaluated, and produces better forecasts then WRF without data assimilation. The thermodynamic analysis showed that the WRF with data assimilation producing vertical profiles of air temperature and dew point temperature very close to the observed profiles. Additional experiments indicate that data assimilated from other sources, in addition to the INMET automatic weather stations and atmospheric soundings, as well as the increases of horizontal resolution in the integration of the WRF with inclusion of subset, provide significant improvements in weather forecasting fields.

Keywords: data assimilation, 3DVAR, WRF.

Introduction

A technique that represents the state of the art in meteorological data assimilation is the Three-Dimensional Variational Method (3DVAR). The first applications of this method in meteorology were held at the European Centre for Medium-Range Weather Forecasts (ECMWF) in the 1990s. Since then the method has been extensively studied and applied [1-8]. The Weather Research and Forecasting Model (WRF) is a Numerical Weather Prediction (NWP), widely used for weather forecasting and research purposes [10]. Currently, several research institutions and government agencies contribute to its development.

The WRF model presents a data assimilation module [2, 6], which is used to perform this work. In this research the data from automatic weather stations from Brazilian National Institute of Meteorology (INMET), and atmospheric sounding data, available from the website of the University of Wyoming, are assimilated to the domain corresponding to South of Brazil. In addition the data available from the Global Telecommunication System (GTS) are also assimilated. Global Telecommunication System is the core of the World Meteorological Organization (WMO) responsible for providing meteorological data from its member countries.

The aim of this study is to evaluate the results of the data assimilation of INMET automatic stations and atmospheric sounding stations, generated by the 3DVAR method in the WRF model prediction (WRF-3DVAR). This review is proceeded by checking if the assimilation with WRF-3DVAR reproduces the observed synoptic scenario and produces better predictions than the model without assimilation procedure. The evaluation is done in a subjective way by comparing the forecasts produced by the model with data assimilation

(WRF-3DVAR) to the forecasts produced by the model without data assimilation (WRF). The specific aim is to evaluate the assimilation procedure in predicting of two rainfall events occurred in 2012 in South of Brazil.

The study is especially important, since the data of the INMET automatic weather stations are not broadcasted by the GTS. Therefore, these data are not assimilated in the forecasts generated by the Global Forecast System (GFS), which provides the initial and boundary conditions for regional models, such as WRF. Moreover, the spatial resolution of WRF model (18 km) is higher than the GFS (50 km). In order to evaluate the performance of the WRF-3DVAR to southern of Brazil, were analyzed meteorological fields and thermodynamic profiles of the atmosphere.

1. Material and methods

The INMET weather station network supplies data to WMO's World Weather Watch program via the international GTS network providing information to national meteorological institutes worldwide. However, the data from INMET, used in this study, are not broadcast by GTS. Therefore, these data are not assimilated by global model which provides the initial and boundary conditions for WRF. In additional experiment SYNOP, METAR, SHIP and BUOY data available to WMO members by GTS were used.

The conversion of INMET weather station and sounding data from ASCII to LITTLE_R format (format required by the data assimilation system of WRF) is done by script wrote in FORTRAN programming language, specially developed for this purpose.

Numerical model is integrated using 35 layers vertically in hybrid sigma-eta coordinate, horizontal domain of 188 by 112 grid points in the east and north directions, in Lambert projection for two spatial resolution: the coarser domain for 18 km in hydrostatic mode and nested domain for 6 km in non-hydrostatic mode (time-step 60 s). The boundary layer processes are evaluated using the level 2 closure according to Mellor and Yamada [11] and the soil model was formulated as five stratified layers. Precipitation process considers the grid scale scheme including parameterized cloud microphysics. Sea surface temperature, fixed during integration, is obtained from monthly climatic data. During assimilation procedure a 2 hours time-window is applied, centered in the analysis time.

The cases elected to be simulated in this research, include two rainfall events occurring in south of Brazil, caused by a cyclogenesis associated to a cold front with strong baroclinic instability. The first case has generated large amounts of rainfall associated with cyclogenesis on September 18, 2012 and occlusion on September 19, 2012 (Case 1); and the second case has generated rainfall amounts associated with cyclogenesis on November 22, 2012 and occlusion in day November 24, 2012 (Case 2).

The variables analyzed are: 6-hours Precipitation Totals (RAIN6), 12-hour Precipitation Totals (RAIN12), Reduction of Pressure to Sea Level (PRES) and vertical profiles of Surface Air Temperature (TC) and the Dew Point Temperature (TD).

2. Results and discussion

It will next be shown the results obtained from the processing of WRF and WRF-3DVAR for the two cases. Regarding to location, RAIN6 forecasted by WRF and WRF-3DVAR to Case 1, were underestimated and poorly reproduced. It was showed a delay in the precipitating core forecasted by WRF-3DVAR comparing to WRF.

In case 2, is shown a significant improvement in the amount and localization of rainfall forecasted by WRF-3DVAR comparing to WRF. The ground truth are 82 rain gauge from INMET. The Fig. 1 shows the RAIN6 predicted from WRF-3DVAR to midnight on November 24, 2012.

The large errors in RAIN6 forecasted by both, WRF and WRF-3DVAR, led the authors to investigate the RAIN12. For case 1, the results were similar to those obtained for RAIN6. Otherwise, for the second case, the amount and location of RAIN12 provided by the WRF-3DVAR were quite well reproduced. The Fig. 2 shows the RAIN12 predicted from WRF-3DVAR to midnight on November 24, 2012.

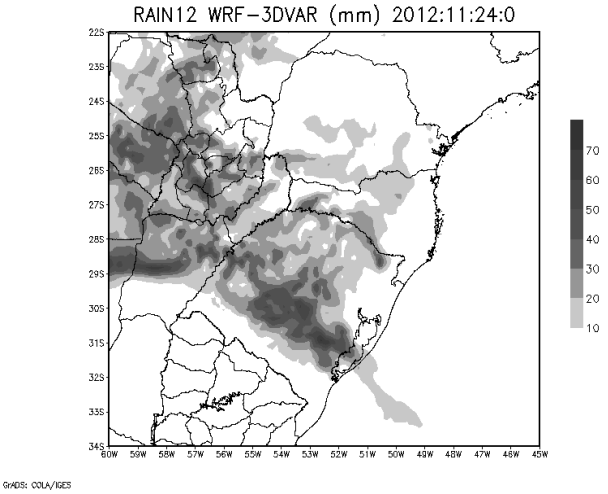


Fig. 2. RAIN12 predicted from WRF-3DVAR to midnight on November 24, 2012

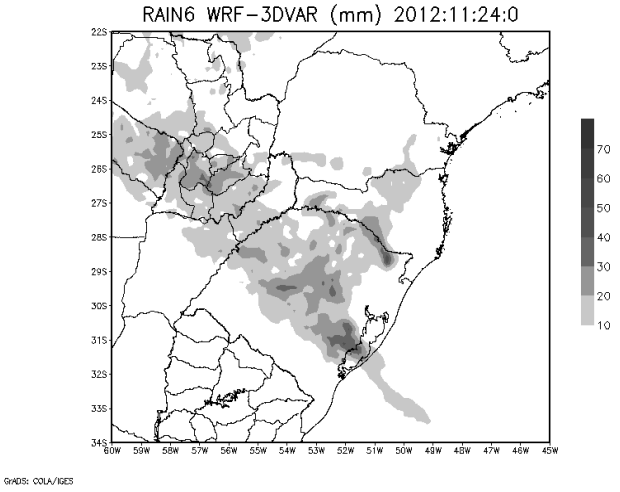


Fig. 1. RAIN6 predicted from WRF-3DVAR to midnight on November 24, 2012

A second experiment was performed in order to verify the results of assimilation by WRF-3DVAR from other source of data (144 SYNOP, 77 METAR, 8 SHIP and 2 BUOY). It

was carry out for nested grids. A nest is a finer-resolution model run. It is embedded simultaneously within a coarser-resolution model run. In this case, the parental grid has 18 km, covering the entire Southern Region of Brazil and the nested grid has 6 km covering only the state of Rio Grande do Sul. The large amounts of RAIN12 near to Foz do Iguaçu city, generated by assimilation of additional data, are in agreement with ground truth.

Table 1

RAIN12 observed from INMET stations

CITY	RAIN12 observed 00:00 UTC 24/11/2012, mm
Porto Alegre	0.0
Santa Maria	36.8
Rio Grande	0.0
Joaçaba	2.2
Florianópolis	1.6
Foz do Iguaçu	40.2
Curitiba	0.0

It is noteworthy that the location of some core of rainfall on the border between Rio Grande do Sul and Uruguay and in some cities of Rio Grande do Sul (Table 1) and Uruguay were better represented in the experiment with high spatial resolution. The surface pressure tendency is an indication of the direction and development of a cyclonic disturbance and it is associated to an increase of rainfall rates in low pressure area. The results show that this scenario was well represented by WRF-3DVAR in Case 1 as well Case 2.

Regarding the vertical profiles of air temperature, TC and TD predicted by WRF-3DVAR are quite close to those observed in Curitiba city above 200 hPa and from surface to 500hPa in Porto Alegre city in Case 1; from surface to 500 hPa in Curitiba city, Foz do Iguaçu city and Florianópolis city; and near the surface in Porto Alegre city in Case 2.

Conclusion

The WRF-3DVAR satisfactorily reproduced the synoptic scenario in the two cases analyzed and the forecasting produced by WRF-3DVAR is markedly better than this one produced by WRF, in the cases studied. However, RAIN6 was poorly reproduced by the model with and without data assimilation in Case 1, duo the spin up problem [9, 10]. The data assimilated from other sources, in addition to the INMET automatic weather stations and soundings stations, as well as the increases of horizontal resolution, provides significant improvements in weather forecasting fields.

This work was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil.

References

1. Rabier, F. Recent Experiments on 4D-Var and First Results from a Simplified Kalman Filter // F. Rabier, J-F. Mahfouf, M. Fischer, H. Järvinen, A. Simmon, F. Bouttier, P. Coutier, M. Hamrud, J. Haseler, A. Hollingsworth, L. Isaksen,

E. Klinker, S. Saarinen, C. Temperton, J-N. Thépaut, P. Undén, D. Vasiljevic // *ECMWF Newsletter.* – 1998. – Vol. 81. – P. 8-16.

2. Barker, D. M. A Three-Dimension (3DVAR) Data Assimilation System For Use With MM5: Implementation and Initial Results / D.M. Barker, W. Huang, Y.-R. Guo, Q. N. Xiao // *Monthly Weather Review.* – 2004. – Vol. 132. – P. 897-914.
3. Meng, Z. Tests of an Ensemble Kalman Filter for Mesoscale and Regional-scale Data Assimilation. Part III: Comparison with 3DVAR in a Real-data Case Study / Z. Meng, F. Zhang // *Monthly Weather Review.* – 2008. – Vol. 136. – P. 522-540.
4. Sugimoto, S. An Examination of WRF 3DVAR Radar Data Assimilation on Its Capability in Retrieving Unobserved Variables and Forecasting Precipitation through Observing System Simulation Experiments / S. Sugimoto, N.A. Crook, J. Sun, Q. Xiao, D.M. Barker // *Monthly Weather Review.* – 2009. – Vol. 137. – P. 4011-4029.
5. Rakesh, V. Intercomparison of the performance of MM5/WRF with and without data assimilation in short-range forecast applications over the Indian region / V. Rakesh, R. Singh, P.C. Joshi // *Meteorology and Atmospherical Physics.* – 2009. – Vol. 105. – P. 133-155.
6. Barker, D.M. The weather Research and Forecasting Model's Community Variational / Ensemble Data Assimilation System: WRFDA / D.M. Barker, W. Huang, Z. Liu, T. Auligné, X. Zhang, S. Rugg, R. Ajjaji, A. Bourgeois, J. Bray, Y. Chen, M. Demirtas, Y.-R. Guo, T. Henderson, W. Huang, H.-C. Lin, J. Michalakes, S. Rizvi, X. Zhang // *Bulletin of American Meteorological Society.* – 2012. – Vol. 93. – P. 831-843.
7. Huang, X.-Y. Four-Dimensional Variational Data Assimilation for WRF: Formulation and Preliminary Results / X.-Y. Huang, Q. Xiao, D.M. Barker, X. Zhang, J. Michalakes, W. Huang, T. Henderson, J. Bray, Y. Chen, Z. Ma, J. Dudhia, Y. Guo, X. Zhang, D.-J. Won, H.-C. Lin, Y. H. Kuo // *Monthley Weather Review.* – 2009. –Vol. 137. – P. 299-314.
8. Routray, A. Impact of Doppler weather radar data on numerical forecast of Indian monsoon depressions / A. Routray, U.C. Mohanty, S.R.H. Rizvi, D. Niyogi, K. K. Osuri, D. Pradhan // *Quarterly Journal of Royal Meteorological Society.* – 2010. – Vol.136. – P. 1836-1850.
9. Oyama, M.D. Estudo Preliminar sobre o tempo de "Spin Up" da umidade do solo no modelo climático do CPTEC / M.D. Oyama, E.J.P. Rocha, C.A. Nobre // *In Proceedings: XI Congresso Brasileiro de Meteorologia, 16-20 de Outubro, 2000, Rio de Janeiro.* – P. 1085-1094.
10. Wang, W. WRF-ARW V3: User's Guide / W. Wang, C. Beezley, M. Duda, et al. URL: http://www.mmm.ucar.edu/wrf/users (accessed: 11.01.2013).
11. Mellor, G.L. A hierarchy of turbulent closure models for planetary boundary layers / G.L. Mellor, T.Yamada // *Journal of Atmospheric Science.* – 1974. – Vol. 31 – P. 1791-1806.

Vinicius Carvalho Beck, student of Graduate Program in Meteorology, Pelotas Federal University (Pelotas, RS, Brazil), vonoco@gmail.com.

Yoshihiro Yamasaki, professor, Faculty of Meteorology, Pelotas Federal University (Pelotas, RS, Brazil), yamasaki@fis.ua.pt.

Fabricio Pereira Harter, professor, Faculty of Meteorology, Pelotas Federal University (Pelotas, RS, Brazil), fabricio.harter@ufpel.edu.br.

Received 11 October 2013.

Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 2, pp. 122–129

УДК 551.509, 004.94

ВЛИЯНИЕ АССИМИЛЯЦИИ ДАННЫХ WRF-3DVAR НА ПРОГНОЗ ЛИВНЕВЫХ ОСАДКОВ НАД ЮЖНОЙ ЧАСТЬЮ БРАЗИЛИИ

В.К.Бек, И. Ямазаки, Ф.П. Хартер

Рассмотрена процедура подачи на вход математической модели данных, сопровождаемых шумом, с целью улучшения статистическими методами моделирования метеорологических полей при прогнозировании погоды. Этот процесс, называемый ассимиляцией данных, является передовым методом в моделировании метеорологических полей. Подход 3DVAR, применяемый в представленном исследовании, является современной технологией ассимиляции данных. Поставлена цель оценить результаты ассимиляции данных сети автоматических станций и данных атмосферного зондирования Бразильского национального метеорологического института (INMET) для прогноза погоды с помощью модели WRF. Регион исследования – южная часть Бразилии. В частности, рассматриваются два события 2012 г., связанные с интенсивными осадками. Представленное исследование важно, поскольку данные автоматических станций INMET не поступают в глобальную телекоммуникационную систему и, следовательно, не используются при производстве прогноза глобальными прогностическими моделями, такими как GFS, которые рассчитывают начальные и граничные условия для региональных моделей, например, WRF. Результаты исследования показывают, что модель WRF с использованием ассимиляции данных в обоих рассмотренных случаях удовлетворительно воспроизводит синоптическую ситуацию, предоставляемую глобальной моделью GFS, и это воспроизведение лучше, чем без использования ассимиляции данных. Термодинамический анализ демонстрирует, что WRF с использованием ассимиляции данных воспроизводит вертикальные профили температуры и точки росы очень близко к наблюдаемым. Дополнительные эксперименты показывают, что усваиваемые данные из других источников в дополнении к данным INMET, а также увеличение пространственного разрешения по горизонтали при интегрировании в WRF с включением дополнительных наборов приводит к значительному улучшению прогнозируемых полей метеорологических величин.

Ключевые слова: ассимиляция данных, 3DVAR, WRF.

Литература

1. Rabier, F. Recent Experiments on 4D-Var and First Results from a Simplified Kalman Filter // F. Rabier, J-F. Mahfouf, M. Fischer, H. Järvinen, A. Simmon,

F. Bouttier, P. Coutier, M. Hamrud, J. Haseler, A. Hollingsworth, L. Isaksen, E. Klinker, S. Saarinen, C. Temperton, J-N. Thépaut, P. Undén, D. Vasiljevic // *ECMWF Newsletter*. – 1998. – Vol. 81. – P. 8-16.

2. Barker, D. M. A Three-Dimension (3DVAR) Data Assimilation System For Use With MM5: Implementation and Initial Results / D.M. Barker, W. Huang, Y.-R. Guo, Q. N. Xiao // *Monthly Weather Review.* – 2004. – Vol. 132. – P. 897-914.
3. Meng, Z. Tests of an Ensemble Kalman Filter for Mesoscale and Regional-scale Data Assimilation. Part III: Comparison with 3DVAR in a Real-data Case Study / Z. Meng, F. Zhang // *Monthly Weather Review.* – 2008. – Vol. 136. – P. 522-540.
4. Sugimoto, S. An Examination of WRF 3DVAR Radar Data Assimilation on Its Capability in Retrieving Unobserved Variables and Forecasting Precipitation through Observing System Simulation Experiments / S. Sugimoto, N.A. Crook, J. Sun, Q. Xiao, D.M. Barker // *Monthly Weather Review.* – 2009. – Vol. 137. – P. 4011-4029.
5. Rakesh, V. Intercomparison of the performance of MM5/WRF with and without data assimilation in short-range forecast applications over the Indian region / V. Rakesh, R. Singh, P.C. Joshi // *Meteorology and Atmospherical Physics*. – 2009. – Vol. 105. – P. 133-155.
6. Barker, D.M. The weather Research and Forecasting Model's Community Variational / Ensemble Data Assimilation System: WRFDA / D.M. Barker, W. Huang, Z. Liu, T. Auligné, X. Zhang, S. Rugg, R. Ajjaji, A. Bourgeois, J. Bray, Y. Chen, M. Demirtas, Y.-R. Guo, T. Henderson, W. Huang, H.-C. Lin, J. Michalakes, S. Rizvi, X. Zhang // *Bulletin of American Meteorological Society.* – 2012. – Vol. 93. – P. 831-843.
7. Huang, X.-Y. Four-Dimensional Variational Data Assimilation for WRF: Formulation and Preliminary Results / X.-Y. Huang, Q. Xiao, D.M. Barker, X. Zhang, J. Michalakes, W. Huang, T. Henderson, J. Bray, Y. Chen, Z. Ma, J. Dudhia, Y. Guo, X. Zhang, D.-J. Won, H.-C. Lin, Y. H. Kuo // *Monthley Weather Review.* – 2009. –Vol. 137. – P. 299-314.
8. Routray, A. Impact of Doppler weather radar data on numerical forecast of Indian monsoon depressions / A. Routray, U.C. Mohanty, S.R.H. Rizvi, D. Niyogi, K. K. Osuri, D. Pradhan // *Quarterly Journal of Royal Meteorological Society.* – 2010. – Vol.136. – P. 1836-1850.
9. Oyama, M.D. Estudo Preliminar sobre o tempo de "Spin Up" da umidade do solo no modelo climático do CPTEC / M.D. Oyama, E.J.P. Rocha, C.A. Nobre // *In Proceedings: XI Congresso Brasileiro de Meteorologia, 16-20 de Outubro, 2000, Rio de Janeiro.* – P. 1085-1094.
10. Wang, W. WRF-ARW V3: User's Guide / W. Wang, C. Beezley, M. Duda, et al. URL: http://www.mmm.ucar.edu/wrf/users (accessed: 11.01.2013).
11. Mellor, G.L. A hierarchy of turbulent closure models for planetary boundary layers / G.L. Mellor, T.Yamada // *Journal of Atmospheric Science.* – 1974. – Vol. 31 – P. 1791-1806.

Винисиус Карвальо Бек, магистрант метеорологического факультета, Федеральный университет г. Пелотас, Бразилия.

Йошихиро Ямазаки, преподаватель метеорологического факультета, Федеральный университет г. Пелотас, Бразилия.

Фабрисио Перейра Хартер, преподаватель метеорологического факультета, Федеральный университет г. Пелотас, Бразилия.

Поступила в редакцию 11 октября 2013 г.

СВЕДЕНИЯ ОБ ИЗДАНИИ

Серия основана в 2012 году.

Свидетельство о регистрации ПИ ФС77-57377 выдано 24 марта 2014 г. Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций.

Журнал включен в Реферативный журнал и Базы данных ВИНИТИ. Сведения о журнале ежегодно публикуются в международной справочной системе по периодическим и продолжающимся изданиям «Ulrich's Periodicals Directory».

ПРАВИЛА ДЛЯ АВТОРОВ

1. Правила подготовки рукописей и пример оформления статей можно загрузить с сайта серии `http://vestnikvmi.susu.ru`. **Статьи, оформленные без соблюдения правил, к рассмотрению не принимаются и назад авторам не высылаются.**
2. Адрес редакции научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»:

   Россия 454080, г. Челябинск, пр. им. В.И. Ленина, 76, Южно-Уральский государственный университет, факультет Вычислительной математики и информатики, кафедра СП, ответственному секретарю, доценту Цымблеру Михаилу Леонидовичу.
3. Адрес электронной почты редакции: `vestnikvmi@gmail.com`
4. **Плата с авторов за публикацию рукописей не взимается, и гонорары авторам не выплачиваются.**
5. Подписной индекс научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»: 10244, каталог «Пресса России». Периодичность выхода — 4 выпуска в год (февраль, май, август и ноябрь).