



ВЕСТНИК

ЮЖНО-УРАЛЬСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА

2014
Т. 3, № 1

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 2014
Vol. 3, no. 1

SERIES

“COMPUTATIONAL
MATHEMATICS AND SOFTWARE
ENGINEERING”

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 Council

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

АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ ПРОГРАММ ДЛЯ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ ПО НЕПРОЦЕДУРНЫМ СПЕЦИФИКАЦИЯМ А.Н. Андрианов, А.Б. Бугеря, Е.Н. Гладкова, К.Н. Ефимкин, П.И. Колударов	5
АЛГОРИТМЫ КРАТКОСРОЧНОГО ПРОГНОЗА С ИСПОЛЬЗОВАНИЕМ РАДИОЛОКАЦИОННЫХ ДАННЫХ: ОЦЕНКА ТРАНСЛЯЦИИ И КОМПОЗИЦИОННЫЙ ДИСПЛЕЙ ЖИЗНЕННОГО ЦИКЛА С.М. Абдуллаев, О.Ю. Ленская, А.О. Гаязова, О.Н. Иванова, А.А. Носков, Д.Н. Соболев, Г.И. Радченко	17
МОДЕЛИРОВАНИЕ СЕЙСМИЧЕСКОГО ПОЛЯ В АКУСТИЧЕСКОМ ПРИБЛИЖЕНИИ ДВУХФАЗНЫХ, ИЕРАРХИЧЕСКИ НЕОДНОРОДНЫХ СРЕД О.А. Хачай, А.Ю. Хачай	33
РАЗРАБОТКА СИСТЕМЫ ДИНАМИЧЕСКОГО ИЗМЕНЕНИЯ КОНТЕНТА ВИДЕОИГР НА ОСНОВЕ ЭВОЛЮЦИОННОГО МОДЕЛИРОВАНИЯ М.Г. Меженин	44
ОПТИМИЗАЦИЯ МОДЕЛИРОВАНИЯ БЕЛКОВЫХ ВЗАИМОДЕЙСТВИЙ НА МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ С ПРЕДОСТАВЛЕНИЕМ ДОСТУПА К АЛГОРИТМУ ЧЕРЕЗ ВЕБ-ИНТЕРФЕЙС К.В. Романенков, А.Н. Сальников	55
О НЕКОТОРЫХ ПРОБЛЕМАХ «ЗАОБЛАЧНОГО» МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ В.П. Ильин	68
МОДЕЛИРОВАНИЕ ЭКСТРЕМАЛЬНОГО НАВОДНЕНИЯ В ДЕЛЬТЕ ДОНА НА МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ В.Н. Дацюк, Л.А. Крукиер, А.Л. Чикин, Л.Г. Чикина	80
МОДЕЛИРОВАНИЕ НА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ ПРОЦЕССОВ РАСПРОСТРАНЕНИЯ МНОГОКОМПОНЕНТНЫХ ПРИМЕСЕЙ В ВОДОЕМЕ И.Н. Шабас	89
ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧ ДИНАМИКИ ЗАРЯЖЕННЫХ ЧАСТИЦ С УЧЕТОМ БАЛАНСИРОВКИ ВЫЧИСЛИТЕЛЬНОЙ НАГРУЗКИ Е.А. Берендеев, М.А. Боронина, В.Д. Корнеев	97
Краткие сообщения	
МЕТОД РАСПРЕДЕЛЕННОГО ОБНАРУЖЕНИЯ ИЗМЕНЕНИЯ КОНЦЕПЦИИ А.А. Волков, Л. Бюх, А. Андреев	113

Contents

AUTOMATIC PROGRAMS GENERATING FROM NON-PROCEDURAL SPECIFICATIONS FOR GPUS A.N. Andrianov, A.B. Bugerya, E.N. Gladkova, K.N. Efimkin, P.I. Koludarov	5
SHORT-RANGE FORECASTING ALGORITHMS USING RADAR DATA: TRANSLATION ESTIMATE AND LIFE-CYCLE COMPOSITE DISPLAY S.M. Abdoulaev, O.U. Lenskaia, A.O. Gayazova, O.N. Ivanova, A.A. Noskov, D.N. Sobolev, G.I. Radchenko	17
MODELING OF SEISMIC FIELDS IN HIERARCHIC HETEROGENEOUS TWO PHASE MEDIA O.A. Hachay, A.Y. Khachay	33
EVOLUTIONARY-BASED PROCEDURAL CONTENT GENERATION FOR VIDEOGAMES M.G. Mezhenin	44
OPTIMIZATION OF MODELING IMMOBILIZED PROTEIN INTERACTIONS ON COMPUTATIONAL CLUSTERS WITH THE SUPPLYING OF ACCESS TO THE ALGORITHM VIA WEB-INTERFACE K.V. Romanenkov, A.N. Salnikov	55
ON SOME "OVER-CLOUD" PROBLEMS OF MATHEMATICAL MODELLING V.P. Il'in	68
MODELING OF EXTREME FLOODS IN THE DELTA OF DON RIVER ON THE MULTIPROCESSOR COMPUTER SYSTEMS V.L. Datsuk, L.A. Krukier, A.L. Chikin, L.G. Chikina	80
SIMULATION OF MULTICOMPONENT POLLUTION DISTRIBUTION IN RESERVOIR ON HIGH PERFORMANCE COMPUTING SYSTEMS I.N. Shabas	89
PARALLEL ALGORITHM FOR SOLUTION OF PROBLEMS OF CHARGED PARTICLE DYNAMICS BY THE USE OF LOAD BALANCE E.A. Berendeev, M.A. Boronina, V.D. Korneev	97
Brief Reports	
A METHOD FOR DISTRIBUTED CONCEPT DRIFT DETECTION A.A. Volkov, L. Büch, A. Andrzejak	113

АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ ПРОГРАММ ДЛЯ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ ПО НЕПРОЦЕДУРНЫМ СПЕЦИФИКАЦИЯМ¹

*А.Н. Андрианов, А.Б. Бугеря, Е.Н. Гладкова, К.Н. Ефимкин,
П.И. Колударов*

Рассмотрены методы и средства автоматического построения параллельных программ для графических процессоров по непроцедурной спецификации задачи, заданной на языке Норма. Приводятся алгоритмы автоматического построения программ для графических процессоров, основанные на анализе зависимостей и выявлении внутреннего параллелизма графа потока управления, полученного по программе на языке Норма. Приведены результаты применения описанных алгоритмов для создания расчетной программы с использованием технологии CUDA при решении задачи из области газодинамики и оценена производительность получающихся параллельных программ.

Ключевые слова: суперкомпьютеры, параллельное программирование, графические процессоры, CUDA, Норма, непроцедурные спецификации.

Введение

В современном научном сообществе задача разработки эффективных параллельных программ имеет важное стратегическое значение. Наверное, уже не осталось ни одной области науки или отрасли промышленности, так или иначе не связанной со сферой высокопроизводительных вычислений. Несмотря на то, что параллельное программирование появилось уже достаточно давно, успешно развивается и проводится масса исследований на эту тему, вопрос «как создать эффективную параллельную программу для решения такой-то задачи» до сих пор крайне актуален для программистов и математиков.

Достаточно быстрое развитие новых аппаратных возможностей для поддержки параллельных вычислений, наблюдаемое в последнее время, еще больше усложняет проблему. Например, появление массово доступных многоядерных процессоров поставило вопрос об эффективном программировании для них. Практически одновременно появились массово доступные графические ускорители (графические процессоры), и опять возник вопрос об эффективном программировании для них. Агрессивное продвижение своих решений фирмами-производителями вычислительных систем, обладающих этими возможностями, часто дезориентирует прикладных специалистов, разрабатывающих параллельные вычислительные программы, толкает их на изменение средств разработки программ, хотя ясные и достаточно убедительные аргументы в пользу таких изменений отсутствуют. Так, применение технологии CUDA для эффективного программирования для графических ускорителей в первых своих версиях являлось, фактически, программированием на уровне ассемблера, с учетом тонких особенностей аппаратуры.

¹ Статья рекомендована к публикации программным комитетом Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: все грани параллелизма – 2013».

С помощью такого ручного низкоуровневого программирования за последние годы некоторые расчетные прикладные пакеты и математические библиотеки были портированы для использования на вычислительных системах с графическими процессорами [1]. Это, несомненно, существенно облегчает задачу прикладному специалисту, но только в том случае, если всего его потребности в высокопроизводительных вычислениях покрываются имеющимися распараллеленным пакетом и/или библиотеками. Если же с помощью таких готовых средств построить решение для своей задачи не удастся, то иного пути, кроме как изучать CUDA или ATI Stream (или OpenCL – это уже более высокий уровень абстракции, но и эффективность его реализации пока далека от желаемого уровня) и начать реализовывать свой алгоритм на столь низком уровне, у прикладного специалиста нет.

Надежды на автоматическое распараллеливание уже написанных последовательных программ на графические процессоры пока совершенно не оправдываются, несмотря на то, что фирмы-производители графических процессоров, как NVIDIA, так и AMD, активно поддерживают данное направление исследований. Из уже реализованных подходов можно отметить те, которые базируются на вполне разумном симбиозе распараллеливающего компилятора и подсказок со стороны программиста, выполненных в виде специальных программных директив, например OpenACC [2]. Но, к сожалению, производительность получаемого таким путем программного кода далека от производительности программ, написанных ручным программированием для целевой платформы, и поэтому пока не может быть признанной удовлетворительной.

Работы по созданию и продвижению новых средств и языков программирования для графических процессоров, гибридных решений и различных нетрадиционных вычислительных архитектур (например, FPGA), ведутся весьма активно [3, 4], однако проблема простой разработки параллельных программ и утилизации новых возможностей вычислительной техники так и остается в настоящее время не решенной.

В данной статье рассмотрены методы и средства автоматического построения параллельных программ для графических процессоров по непроцедурной спецификации на примере решения этой задачи для спецификаций, заданных на языке Норма. В разделе 1 кратко рассказывается о языке Норма и его основных свойствах. Во втором разделе сформулированы методы автоматического построения программы с использованием технологии CUDA по непроцедурным спецификациям, заданным на языке Норма. В разделе 3 приведен пример построения такой программы в соответствии с предложенными методами. В разделе 4 рассказано об опыте применения предложенных методов для создания расчетной программы с использованием технологии CUDA при решении задачи из области газодинамики и оценена производительность получающихся параллельных программ. В заключении говорится о продолжающихся работах коллектива авторов по данной тематике.

1. Декларативный подход. Язык Норма

Один из возможных подходов к решению задачи автоматизации параллельного программирования вычислительных задач, и, в частности, задачи автоматического построения эффективной программы для графических процессоров, является подход с использованием декларативных (непроцедурных) языков. При использовании этого подхода прикладной специалист программирует решение вычислительной задачи на непроцедур-

ном языке (понятия, связанные с архитектурой параллельного компьютера, моделями параллелизма и др. при этом не используются), а затем компилятор автоматически строит параллельную программу (учитывая архитектуру целевого параллельного компьютера, модели параллелизма и прочее.). С учетом отмеченных выше проблем привлекательность этого подхода в настоящее время только усиливается и интерес к идеям непроцедурного декларативного программирования и реализации этих идей в языках программирования неуклонно растет.

Идеи декларативного программирования были сформулированы еще в прошлом веке, теоретические исследования этого подхода для класса вычислительных задач проведены в пионерских работах И.Б. Задыхайло [5] еще в 1963 г. Непроцедурный язык Норма и система программирования Норма [6-8] разработаны в ИПМ им. М.В. Келдыша РАН также достаточно давно и предназначены для автоматизации решения вычислительных сеточных задач на параллельных компьютерах. Расчетные формулы записываются на языке Норма в математическом, привычном для прикладного специалиста виде. Язык Норма позволяет описывать решение широкого класса задач математической физики. Программа на языке Норма имеет очень высокий уровень абстракции и отражает метод решения, а не его реализацию при конкретных условиях. Такое описание не ориентировано на конкретную архитектуру компьютера, поэтому оно предоставляет большие возможности для выявления естественного параллелизма и организации вычислений.

В настоящее время в ИПМ им. М.В. Келдыша ведутся работы по созданию версии компилятора программ на языке Норма+, который на выходе создает исполняемую программу для графических процессоров фирмы NVIDIA с использованием технологии CUDA.

2. Принципы построения CUDA программы по декларативным описаниям

При трансляции с языка Норма+ решается задача синтеза выходной параллельной программы, то есть выходная параллельная программа строится автоматически. В результате анализа зависимостей по данным между операторами программы на языке Норма+, в случае разрешимости этих зависимостей, представляется возможным построить так называемую «параллельную ярусную схему» выполнения программы. На каждом ярусе данной ярусной схемы располагаются операторы программы, которые не имеют зависимостей друг от друга и могут выполняться независимо и, соответственно, параллельно. В то же время каждый из этих операторов имеет зависимость от одного или более операторов, располагающихся на предыдущем ярусе ярусной схемы. Таким образом, группу операторов, располагающихся на одном уровне ярусной схемы, можно выполнять параллельно в результирующей программе, но только после того, как будут полностью выполнены все операторы предыдущего уровня ярусной схемы. Параллельная ярусная схема программы является, фактически, представлением идеального (естественного) параллелизма, определяемого соотношениями и зависимостями между расчетными переменными программы.

В результате ряда исследований по трансформации описанной параллельной ярусной схемы программы на языке Норма+ в программу с использованием технологии NVIDIA CUDA для графических процессоров авторами предлагается использовать сле-

дующий подход для автоматического построения исполняемой программы для графических процессоров с использованием технологии CUDA.

Исполняемая программа стартует и завершается на центральном процессоре, на нем же выполняется ввод-вывод данных и итерационные циклы. Очевидно также, что вся логика по управлению вычислениями, выделением памяти на графическом процессоре, организация обменов данными между памятью графического и центрального процессора, тоже создается в программном коде, выполняющемся на центральном процессоре. Сами вычислительные операторы выполняются на графическом процессоре. Для этого они группируются в определенные наборы, каждый из которых может выполняться в пределах одного ядра программы с использованием технологии NVIDIA CUDA. Программа, выполняющаяся на центральном процессоре, осуществляет контроль за общим выполнением программы, запускает полученные ядра в определенном порядке, ведет итерационные циклы и проверяет условие выхода из итерации. На рис.1 приведен пример общей схемы выполнения исполняемой программы.

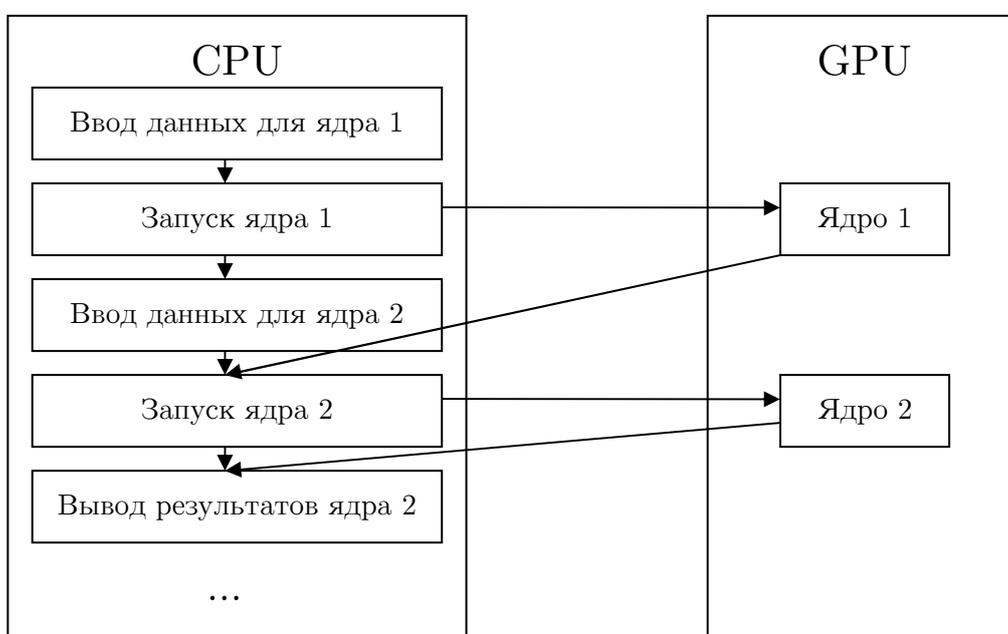


Рис. 1. Пример общей схемы выполнения исполняемой программы

Один из ключевых моментов в этой схеме организации исполняемой программы, и он же алгоритмически самый тяжелый – это задача группировки вычислительных операторов в ядра, которые будут выполняться на графическом процессоре. С одной стороны, чем больше операторов будет содержать такое ядро, т.е. чем крупнее будут исполняемые ядра в программе, тем эффективнее она будет работать, т.к. будет меньше запусков ядер, меньше передачи параметров, в том числе и через память центрального процессора и др. Но, с другой стороны, все операторы, сгруппированные в одно ядро, должны работать с одним и тем же распределением рабочих областей на конфигурацию блоков и нитей, определяемую при запуске ядра. И может оказаться, что поиск такого распределения для большой группы операторов, объединенных в одно ядро, может дать худший результат, чем разбиение этой группы операторов на более мелкие отдельные ядра и поиск распределения для каждой группы в отдельности. Кроме того, при поиске

таких группировок необходимо соблюдать ряд обязательных условий, накладываемых ярусной схемой и сущностью группируемых операторов. Например, мы можем менять порядок выполнения операторов, но только в пределах одного яруса. А операторы ввода-вывода должны выполняться на центральном процессоре.

В итоге, после ряда исследований характеристик получающихся программ, предлагается использовать следующий набор эвристик и обязательных условий для решения задачи группировки операторов программы на языке Норма+ в один набор, который может выполняться в пределах одного ядра программы с использованием технологии NVIDIA CUDA:

1. Каждая нить графического процессора осуществляет вычисления в одной точке расчетной области Норма-программы (другими словами, одной точке индексного пространства вычисления).
2. Поскольку операторы, находящиеся на одном ярусе параллельной ярусной схемы, можно выполнять параллельно, они могут выполняться одним ядром CUDA программы, причем в любом порядке. Поэтому в очередном ярусе выбираются операторы, которые выполняются на одной и той же одномерной, двумерной или трехмерной подобласти. Вообще говоря, оператор может выполняться и на области большей размерности, но выбранная для распараллеливания подобласть должна присутствовать в области выполнения оператора. Эти выбранные операторы могут выполняться одним ядром, при условии соблюдения описанных ниже условий.
3. Выбранная для распараллеливания подобласть разбивается на блоки и нити для ядра графического процессора следующим образом:

- если подобласть одномерная, то за число нитей берется предопределенное число, кратное степени двойки (1024 по умолчанию) и равное максимальному количеству нитей в блоке на целевой архитектуре. А для блоков выбирается одномерный массив с учетом ограничений целевой архитектуры на размер массива блоков по одному направлению или двумерный массив таким образом, чтобы произведение количества блоков на количество нитей полностью покрывало выбранную подобласть;

Пусть $D(N)$ – выбранная для распараллеливания подобласть из N точек. Тогда распределение выглядит так:

$D(N), (((N + 1023)/1024) \leq 65535) \Rightarrow$

`kernel<<< ((N + 1023)/1024), 1024 >>>()`

$D(N), (((N + 1023)/1024) > 65535) \Rightarrow$

`kernel<<< dim3(((N + 1023)/1024 + 65534)/65535), 65535), 1024 >>>()`

- если подобласть двумерная, и одно из направлений меньше или равно максимальному количеству нитей в блоке на целевой архитектуре, то этому направлению будут соответствовать нити, количество которых определяется как ближайшая сверху степень двойки. А для второго направления подобласти выбирается одномерный массив блоков, равный размеру этого направления, с учетом ограничений целевой архитектуры на размер массивов блоков по одному направлению, или подбирается двумерный массив блоков таким образом, чтобы общее количество блоков полностью покрывало второе направления подобласти.

$D(N1, N2), (N2 \leq 1024), (N1 \leq 65535) \Rightarrow$

`kernel<<< N1, N2 >>>()`

$D(N1, N2), (N2 \leq 1024), (N1 > 65535) \Rightarrow$

$\text{kernel} \lll \text{dim3}((N1 + 65534)/65535, 65535), N2 \ggg ()$

- если подобласть двумерная, и ни одно из направлений не может быть представлено нитями в блоке на целевой архитектуре, то одному из направлений назначается двумерный массив, одно измерение которого – нити, количество которых определяется как предопределенное число, кратное степени двойки (1024 по умолчанию) и равное максимальному количеству нитей в блоке на целевой архитектуре. А второе измерение этого двумерного массива выбирается из одного из направлений в двумерном массиве блоков, таким образом, чтобы произведение количества блоков по данному направлению на количество нитей полностью покрывало рассматриваемое направление подобласти. Второму направлению подобласти ставится в соответствие второе направление в двумерном массиве блоков.

$D(N1, N2), (N2 > 1024), (N1 \leq 65535) \Rightarrow$

$\text{kernel} \lll \text{dim3}(N1, (N1 + 1023)/1024), 1024 \ggg ()$

- если подобласть трехмерная, и одно из направлений меньше или равно максимальному количеству нитей в блоке на целевой архитектуре (а остальные направления меньше или равны размеру блока), то этому направлению будут соответствовать нити количеством ближайшая сверху степень двойки. А для остальных направлений подобласти ставится в соответствие двумерный массив блоков соответствующей размерности.

$D(N1, N2, N3), (N3 \leq 1024), (N1, N2 \leq 65535) \Rightarrow$

$\text{kernel} \lll \text{dim3}(N1, N2), N3 \ggg ()$

4. Если оператор является оператором ввода-вывода данных, то он должен выполняться на центральном процессоре, и он может выполняться параллельно с запуском ядра. При этом дальнейшее добавление в формируемое ядро операторов со следующего уровня (п.6) становится невозможным.
5. Если в операторе присутствует функция редукции, и область применения данной функции редукции полностью представлена нитями по какому-то индексу, а все остальные индексы из области применения функции редукции (если они есть) не входят в область, распараллеленную в текущем ядре, то этот оператор может быть выполнен ядром без исключений. В противном же случае, если область применения функции редукции имеет индексы, которым соответствует какое-либо направление массива блоков, данное ядро может выполнить функцию редукции только частично, и затем выполнение ядра должно быть прекращено и запущено следующее ядро (или даже последовательно несколько ядер) с другим распределением – так, чтобы в итоге все частичные результаты выполнения функции редукции были представлены исключительно нитями и было возможно получить окончательный результат. Такой оператор должен быть поставлен последним в ядре, и дальнейшее добавление в формируемое ядро операторов со следующего уровня (п.6) становится невозможным.
6. После завершения формирования последнего вычислительного ядра с операторами данного слоя ярусной схемы можно попробовать добавить в это ядро операторы со следующего слоя ярусной схемы при соблюдении двух описанных ниже условий. Это возможно, т.к. ядро выполняет операторы, содержащиеся в нем, последовательно в отдельно взятой точке распараллеленной области. Если зависимости между операторами разных уровней параллельной ярусной схемы простые, без смещения по какому-

либо индексу из области распараллеливания (например $X = \dots$; $R = \text{func}(X)$), то последовательно выполненные действия в каждой точке области дадут правильный результат. И не важно, скажем, что при вычислении R в точке k значение X в точке $k+1$ еще не было вычислено другой нитью – важно только, что значение X в точке k уже вычислено. Затем, после завершения очередного уровня параллельной ярусной схемы, можно начинать включать в текущее ядро операторы со следующего уровня и т.п., пока процесс добавления не закончен по одной из причин, описанной в п. 4-6. Добавление таких операторов возможно при соблюдении следующих условий:

- если на добавляемом слое есть операторы ввода-вывода, они должны быть выполнены на центральном процессоре после завершения выполнения создаваемого ядра, формирование которого должно быть закончено текущим слоем;
- в добавляемых операторах не должно быть использования переменных с индексами из подобласти распределения со смещением (например $R = \text{func}(X[i+1])$). Если такие операторы есть, то они не должны быть добавлены в создаваемое ядро, формирование которого должно быть закончено текущим слоем. С такими оставшимися операторами можно начать формировать следующее ядро.

3. Пример построения программы с использованием CUDA

В качестве примера рассмотрим применение описанной выше схемы для простого фрагмента программы на языке Норма+, см. рис. 2.

```
Oij: (Oj: (j=1..w); Oi: (i=1..v)).
VARIABLE Vij DEFINED ON Oij DOUBLE.
VARIABLE Vsum DEFINED ON Oi DOUBLE.
DOMAIN PARAMETERS v=2000, w=3000.
FOR Oij ASSUME Vij = j+(i-1)*w.
FOR Oi ASSUME Vsum = SUM((Oj)Vij).
```

Рис. 2. Фрагмент программы на языке Норма+

В приведенном фрагменте величине V_{ij} , определенной на двумерной области O_{ij} с индексами i и j , присваиваются начальные значения $j+(i-1)*w$, а затем производится суммирование по индексу j (по области O_j). В данном случае распараллеливается вся область O_{ij} . По индексу j область распределяется на нити и на одну из размерностей блоков, а по индексу i – на вторую размерность блоков (3-й подпункт пункта 3 приведенной выше схемы). Начало оператора суммирования выполняется в первом же ядре, но, т.к. все направление индекса j не покрывается полностью нитями, необходимо создание второго ядра, завершающего суммирование.

Схема распределения области O_{ij} на блоки и нити приведена на рис. 3.

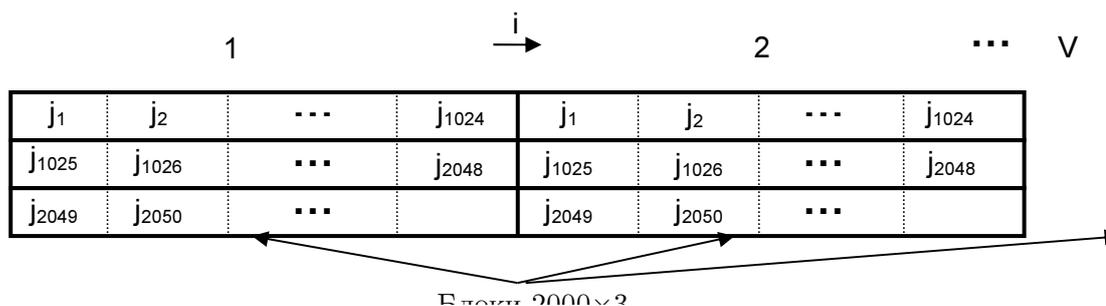


Рис. 3. Отображение области O_{ij} на блоки и нити

Жирными линиями выделены блоки, их конфигурация 2000 (размер по индексу i) на 3 (дополнительное направление по индексу j , которое в объединении с 1024 нитями в каждом блоке обеспечивает представление 3000 точек по индексу j). Нити в каждом блоке показаны пунктиром.

```

#define v 2000
#define w 3000
#define wthreadK1 1024 // Ближайшая сверху степень 2 от w,
                        // с учетом ограничения на кол-во нитей в блоке
#define wBlockYK1 (w+wthreadK1-1)/wthreadK1
#define wthreadK1Red 4 // Ближайшая сверху степень 2 от wBlockYK1

__global__ void SumK1() {
    shared double shared[wthreadK1];
    // Получаем индекс j
    int j = threadIdx.x + blockIdx.y*wthreadK1;
    if (j < w) { // Проверяем, что не вышли за границу области
        int i = blockIdx.x;
        int gidx = i*w + j;
        // Присваиваем начальные значения
        Vij_dev[gidx] = j + 1 + i * w;
        // Начинаем суммирование
        shared[threadIdx.x] = Vij_dev[gidx];
        for (int d = wthreadK1/2; d > 0; d /= 2) {
            int from = threadIdx.x + d;
            __syncthreads();
            if ((threadIdx.x < d) &&
                (from + blockIdx.y*wthreadK1 < w))
                shared[threadIdx.x] += shared[from];
        }
        // Сохраняем частичный результат
        if (threadIdx.x == 0)
            Vsum_block[blockIdx.x*wBlockYK1+blockIdx.y] = shared[0];
    }
}

```

Рис. 4. Программа для первого ядра

На рис. 4 приведена получившаяся программа для первого ядра вместе с общими определениями констант. В ней производится инициализация переменной V_{ij} (в ядре представлена переменной V_{ij_dev}) и начинается ее суммирование по области O_j . Для этого каждый блок вычисляет сумму для всех входящих в него нитей. Но, т.к. точки области O_j распределены на 3 блока, то итоговый результат, который должен быть вычислен как сумма частичных результатов, полученных в каждом из этих 3-х блоков, в рамках данного ядра вычислить невозможно.

Поэтому для завершения суммирования организуется второе ядро. Программа для него приведена на рис. 5.

```

__global__ void SumK1Red() {
    // Осуществляет суммирование Vsum_block
    __shared__ double shared[wBlockYK1];
    if (threadIdx.x < wBlockYK1) {
        int i = blockIdx.x;
        int gidx = i*wBlockYK1 + threadIdx.x;
        // Копируем насчитанные предыдущим ядром частичные данные
        shared[threadIdx.x] = Vsum_block[gidx];
        // Завершаем суммирование
        for (int d = wthreadK1Red/2; d > 0; d /= 2) {
            int from = threadIdx.x + d;
            __syncthreads();
            if ((threadIdx.x < d) && (from < wBlockYK1))
                shared[threadIdx.x] += shared[from];
        }
        // Сохраняем итоговый результат
        if (threadIdx.x == 0)
            Vsum_dev[i] = shared[0];
    }
}

```

Рис. 5. Программа для второго ядра

Фрагмент программы, выполняющейся на центральном процессоре, приведен на рис. 6. Он, как можно заметить, заключается в последовательном вызове ядер на нужной конфигурации блоков и нитей.

```

SumK1<<< dim3(v, wBlockYK1), wthreadK1 >>>();
SumK1Red<<< v, wthreadK1Red >>>();

```

Рис. 6. Фрагмент программы для центрального процессора

4. Предварительные результаты

Приведенная выше схеме построения CUDA программы по декларативным описаниям была применена к программе на языке Норма+ из области газодинамики. Была взята программа на языке Си, полученная компиляцией указанной программы на языке Норма+ в последовательную программу, и ручным способом трансформирована в программу с использованием CUDA так, как это мог бы сделать компилятор по описанной выше схеме. Полученная параллельная программа была запущена на вычислительном кластере К-100 [9]. Производительность параллельной программы сравнивалась с последовательной программой и OpenMP версией той же самой программы. Времена выполнения версий программы приведены в табл.

Таблица

Время выполнения различных версий программы

Последовательная программа, 1 ядро Intel Xeon X5670	OpenMP программа, 11 ядер Intel Xeon X5670	CUDA программа, 1 графический процессор nVidia Fermi C2050
44,5 с	4,62 с	1,35 с

Полученный результат – ускорение в 33 раза по сравнению с последовательной программой – позволяет надеяться на успешное применение схемы для автоматического

построения эффективных программ для графических процессоров по декларативным описаниям.

Заключение

В рамках данной работы были рассмотрены методы и средства автоматического построения параллельных программ для графических процессоров по непроцедурным спецификациям. Приведены алгоритмы автоматического построения программ для графических процессоров, основанные на анализе зависимостей и выявлении внутреннего параллелизма графа потока управления, полученного по непроцедурным спецификациям, заданным на языке Норма. Приведены результаты применения описанных алгоритмов для создания расчетной программы с использованием технологии CUDA при решении задачи из области газодинамики и оценена производительность получающихся параллельных программ.

В настоящее время ведется разработка версии компилятора с языка Норма+ для графических процессоров с использованием технологии CUDA с применением описанной в данной работе схемы.

Работа выполнена при финансовой поддержке гранта РФФИ № 12-01-00527-а.

Литература

1. Кривов, М.А. Опыт портирования среды для HDR-обработки изображений на GPU и APU / М.А. Кривов, М.Н. Притула, С.Г. Елизаров. URL: <http://pavt.susu.ru/2012/short/175.pdf> (дата обращения: 16.08.2013).
2. OpenACC. URL: <http://openacc.org> (дата обращения: 16.08.2013).
3. Бахтин, В.А. Распараллеливание с помощью DVM-системы некоторых приложений гидродинамики для кластеров с графическими процессорами / В.А. Бахтин, И.Г. Бородич, Н.А. Катаев, М.С. Клинов, В.А. Крюков, Н.В. Поддерюгина, М.Н. Притула, Ю.Л. Сазанов // Научный сервис в сети Интернет: поиск новых решений: Труды Международной суперкомпьютерной конференции (17–22 сентября 2012 г., г. Новороссийск). — М.: Изд-во МГУ, 2012. — С. 444–450.
4. Описание языка программирования COLAMO. URL: <http://colamo.parallel.ru> (дата обращения: 16.08.2013).
5. Задыхайло, И.Б. Организация циклического процесса счета по параметрической записи специального вида / И.Б. Задыхайло // Журн. выч. мат. и мат. физ. — 1963. — Т. 3, № 2, — С. 337–357.
6. Андрианов, А.Н. Норма. Описание языка. Рабочий стандарт / А.Н. Андрианов, А.Б. Бугеря, К.Н. Ефимкин, И.Б. Задыхайло — М.: Препринт ИПМ им. М.В. Келдыша РАН, 1995. — № 120. — 52 с.
7. Андрианов, А.Н. Декларативный язык Норма и программирование для новых архитектур: многоядерные системы / А.Н. Андрианов, А.Б. Бугеря, К.Н. Ефимкин, П.И. Колударов // Научный сервис в сети Интернет: Суперкомпьютерные центры и задачи: Труды Международной суперкомпьютерной конференции (20–25 сентября 2010 г., г. Новороссийск). — М.: Изд-во МГУ, 2010. — С. 68–70.
8. Система Норма. URL: <http://www.keldysh.ru/pages/norma> (дата обращения: 16.08.2013).

9. Гибридный вычислительный кластер К-100. URL: <http://www.kiam.ru/MVS/resources/k100.html> (дата обращения: 16.08.2013).

Андрианов Александр Николаевич, д.ф.-м.н., в.н.с., Институт прикладной математики им. М.В. Келдыша РАН (Москва, Российская Федерация), and@a5.kiam.ru.

Бугеря Александр Борисович, к.ф.-м.н., с.н.с., Институт прикладной математики им. М.В. Келдыша РАН (Москва, Российская Федерация), bug@a5.kiam.ru.

Гладкова Екатерина Николаевна, электроник, Институт прикладной математики им. М.В. Келдыша РАН (Москва, Российская Федерация), gladkova@keldysh.ru.

Ефимкин Кирилл Николаевич, к.ф.-м.н., зав. отделом, Институт прикладной математики им. М.В. Келдыша РАН (Москва, Российская Федерация), bigcrocodile@yandex.ru.

Колударов Павел Иванович, стажер-исследователь, Институт прикладной математики им. М.В. Келдыша РАН (Москва, Российская Федерация), koludarov@yandex.ru.

Поступила в редакцию 18 сентября 2013 г.

*Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 1, pp. 5–16*

AUTOMATIC PROGRAMS GENERATING FROM NON-PROCEDURAL SPECIFICATIONS FOR GPUS

A.N. Andrianov, Keldysh Institute of Applied Mathematics RAS (Moscow, Russian Federation),

A.B. Bugerya, Keldysh Institute of Applied Mathematics RAS (Moscow, Russian Federation),

E.N. Gladkova, Keldysh Institute of Applied Mathematics RAS (Moscow, Russian Federation),

K.N. Efimkin, Keldysh Institute of Applied Mathematics RAS (Moscow, Russian Federation),

P.I. Koludarov, Keldysh Institute of Applied Mathematics RAS (Moscow, Russian Federation)

A new approach to automatic programs generating for GPUs from non-procedural task's specification in Norma language is considered. Algorithms of automatic programs generating for GPUs based on dependencies analysis and discovering control flow internal parallelism obtained from Norma program are given. Described algorithms applied to gas-dynamics task solving are carried out in calculating CUDA program. The performance of generated CUDA program is estimated.

Keywords: HPC, parallel programming, GPU, CUDA, Norma language, non-procedural specifications.

References

1. Krivov M.A., Pritula M.N., Elizarov S.G. Opyt portirovaniya sredy dlya HDR- obrabotki izobrazheniy na GPU i APU [Porting HDR Image Processing System to GPU and APU Experience] URL: <http://pavt.susu.ru/2012/short/175.pdf> (accessed: 16.08.2013).
2. OpenACC. URL: <http://openacc.org> (accessed: 16.08.2013).
3. Bakhtin V.A., Borodich I.G., Kataev N.A., Klinov M.S., Krukov V.A., Podderugina N.V., Pritula M.N., Sazanov U.L. Rasparallelivanie s pomoschyu DVM-sistemy nekotorykh prilozheniy gidrodinamiki dlya klasterov s graficheskimi processorami [Parallelization Some Hydrodynamics Applications Using DVM-System for Clusters with GPUs]. Nauchnyy servis v seti Internet: poisk novyh resheniy: Trudy mezhdunarodnoi superkomputernoi konferentsii (17-22 sentyabrya 2012, Novorossiysk) [Scientific Service in Internet: Searching the New Solutions: Proceedings of the International Supercomputing Conference (Novorossiysk, Russia, September, 17-22, 2012)] — Moscow, Publishing of the Moscow State University, 2012. — P. 444–450.
4. Opisaniye yazyka programmirovaniya COLAMO [COLAMO Programming Language Description]. URL: <http://colamo.parallel.ru> (accessed: 16.08.2013).
5. Zadykhailo I.B. Organizatsiya tsiklicheskogo protsessa scheta po parametriceskoy zapisi spetsialnogo vida [Organizing of Cycle Process Calculating by Special Parametrized Record]. Vychislitel'naya matematika i matematicheskaya fizika [Computing Mathematics and Mathematical Physics]. — 1963. — Vol. 3, No 2. — P. 337–357.
6. Andrianov A.N., Bugerya A.B., Efimkin K.N., Zadykhailo I.B. Norma. Opisaniye yazyka. Rabochiy standart [Norma Programming Language. Draft Standard]. — Moscow, Preprint of the Keldysh Institute of Applied Mathematics RAS. — 1995. — No 120. — 52 p.
7. Andrianov A.N., Bugerya A.B., Efimkin K.N., Koludarov P.I. Deklarativniy yazyk Norma i programmirovaniye dlya novyh arkhitektur: mnogoyadernie sistemy [Declarative Norma Language and Programming for New Architectures: Multi-Core Systems]. Nauchnyy servis v seti Internet: Superkomputernie tsentry i zadachi: Trudy mezhdunarodnoi superkomputernoi konferentsii (20-25 sentyabrya 2010, Novorossiysk) [Scientific Service in Internet: Supercomputing Centers and Tasks: Proceedings of the International Supercomputing Conference (Novorossiysk, Russia, September, 20-25, 2010)] — Moscow, Publishing of the Moscow State University, 2010. — P. 68–70.
8. Norma. URL: <http://www.keldysh.ru/pages/norma> (accessed: 16.08.2013).
9. Hybrid Supercomputing K-100. URL: <http://www.kiam.ru/MVS/resourses/k100.html> (accessed: 16.08.2013).

Received 18 September 2013

АЛГОРИТМЫ КРАТКОСРОЧНОГО ПРОГНОЗА С ИСПОЛЬЗОВАНИЕМ РАДИОЛОКАЦИОННЫХ ДАННЫХ: ОЦЕНКА ТРАНСЛЯЦИИ И КОМПОЗИЦИОННЫЙ ДИСПЛЕЙ ЖИЗНЕННОГО ЦИКЛА

*С.М. Абдуллаев, О.Ю. Ленская, А.О. Гаязова, О.Н. Иванова, А.А. Носков,
Д.Н. Соболев, Г.И. Радченко*

В работе представлены алгоритмы обработки радиолокационных данных, необходимые для создания системы сверхкраткосрочного прогноза опасных явлений, включая процедуры определения скорости трансляции, экстраполяционного прогноза и верифицирующего прогноз дисплея жизненного цикла (LCD). Основные особенности этой системы заключаются в следующем. Простая полуавтоматическая процедура смещения консервативных фрагментов поля дает устойчивые оценки трансляции для всей мезомасштабной конвективной системы. В свою очередь, это позволяет экстраполировать текущее поле осадков на сроки до 1 часа и адекватно интерпретировать композиционные изображения LCD в терминах традиционных критериев успешности прогноза. Поскольку LCD отражает области развития и диссипации штормов, а также зоны запрета на возникновения новых очагов конвекции, то в перспективе с его помощью возможна корректировка прогноза и повышение его успешности.

Ключевые слова: мезомасштабные конвективные системы, радиолокационные осадки, алгоритмы движения, прогноз опасных явлений, верификация прогноза, дисплей жизненного цикла.

Введение

Сверхкраткосрочный прогноз погоды на сроки до 6 часов сочетает результаты дистанционных, наземных наблюдений и численных мезомасштабных моделей, и как быстро развивающаяся информационная технология становится повседневным элементом жизни в развитых странах [1]. Главной целью данной работы является разработка алгоритмов прогнозов, позволяющих при минимальном наборе доступных данных, таких как цифровые поля радиолокационных осадков, воспроизводить основные результаты наиболее технически оснащенных и алгоритмически совершенных систем прогноза опасных явлений погоды, описанных в [2, 3]. В частности, нами предложены алгоритмы решения следующих задач систем штормового оповещения: 1) задача адекватного определения движения мезомасштабной конвективной системы (МКС) и 2) задача автоматической верификации экстраполяционного прогноза и диагностики эволюции МКС.

Решение задач основано на том, что движение МКС можно представить как комбинацию двух векторов различной природы: трансляции и эволюции. Трансляция – это не зависящий от масштаба элемента процесс переноса, когда все конвективные ячейки и шторма движутся с одной и той же горизонтальной скоростью, часто называемой «ведущим потоком». В отличие от трансляции, эволюция сочетает процессы появления, развития и диссипации ячеек, штормов и их ансамблей различного масштаба, и потому имеет значительные временные и пространственные вариации. Сопоставление трансляции и эволюции систем различного масштаба и типа [4–11] позволили обосновать комплекс диагно-

стических и прогностических методов [12], многие из которых не потеряли своей актуальности [13–15]. В частности, сопоставление интерактивно определенной трансляции и морфологии поля осадков на стадии максимального развития систем [16], позволяет реконструировать климатологические ряды динамических характеристик, которые в свою очередь являются ядром универсального алгоритма прогноза [12], объединяющего результаты дистанционных наблюдений конвективных систем и результаты их численного прогноза на основе модели WRF [14].

Изложенное выше предполагает, что задачу определения движения МКС по радиолокационным данным следует проводить поэтапно. Статья состоит из введения, двух разделов и заключения.

В первом разделе статьи приводится описание способов определения трансляции МКС, проблемы автоматизации этих способов и возможности разработки интерактивного алгоритма (разделы 1.1–1.3). Второй раздел содержит описание последующих этапов, в течение которых, основываясь на трансляции, производится экстраполяция поля осадков к следующему моменту времени, когда будет доступно новое изображение (раздел 2.1). Далее, сравнивая элементы прогнозного и реального наблюдаемого полей, на третьем этапе можно оценить качество прогноза и, насколько это возможно, выделить эволюционную составляющую движения системы (раздел 2.2). Данные последнего этапа можно использовать для корректировки следующего цикла прогноза. В заключении обобщаются результаты работы.

1. Алгоритмы определения трансляции

В виду очевидной важности процедуры определения трансляции в этом разделе кратко изложены основные проблемы автоматических алгоритмов (п. 1.1), работы проведенные авторами по разработке интерактивных (п. 1.2) и автоматических алгоритмов трансляции (п. 1.3).

1.1. Проблемы автоматических алгоритмов

С алгоритмической точки зрения, скорость трансляции может рассматриваться как скорость адвекции горизонтального поля радарной отражаемости. Предложены различные автоматические алгоритмы оценки последней, которые могут быть классифицированы на структурные и объектные технологии [17, 18].

Расчет матрицы коэффициентов пространственных кросс-корреляций между двумя последовательными дискретными полями осадков – это пример методов, отслеживающих мезомасштабную структуру осадков. Скорость трансляции в этом случае – «наилучший» вектор смещения всего поля.

Алгоритмы идентификации радиолокационных ячеек и их сопровождения – примеры объектного подхода. Скорость трансляции получается осреднением найденных скоростей индивидуальных ячеек. Поскольку ячейки – это наименьшие по размеру (5–10 км) и наиболее короткоживущие элементы МКС (≈ 30 мин), то эти алгоритмы удовлетворительно работают, когда имеются данные с пространственно-временным разрешением не менее $2 \text{ км} \times 2 \text{ км}$ и 10 мин. Часто наблюдается ситуация, когда радар работает в режиме объемного сканирования ≈ 15 мин, и идентификация ячеек в последовательных объемах практически невозможна – в этом случае объектом сопровождения является уже не

ячейки, а штормы, которые в общем случае имеют различные направления эволюции. Поэтому для фильтрации медленно и быстро эволюционирующих элементов поля осадков в современных автоматических алгоритмах применяются различные технологии, сочетающие объектные и структурные подходы [17, 18].

Заметим, что в оперативной практике частота случаев, когда автоматические алгоритмы по той или иной причине не состоятельны, может быть достаточно высокой. Например, в мае-августе 2013 г. при анализе эволюции 132 грозовых МКС, возникших на Европейской территории России и сопредельных территориях Украины и Белоруссии [19], обнаружилось, что предусмотренная автоматизация не смогла определить вектор движения поля в течение более половины времени развития 78 МКС. Анализ 2660 радарных изображений показал, что наибольший процент «отказов» наблюдался в случаях получасовой выдачи информации, при малых скоростях трансляции комплексов локальных штормов ≤ 25 км/ч, в крупных полях осадков слоистообразной облачности с включением конвективных очагов, а также при входе и выходе зон осадков за пределы радиуса обзора радара.

Как мы покажем ниже, интерактивная процедура способна выдать оценку в большинстве этих случаев, когда результаты автоматических методов несостоятельны или трудно интерпретируемы.

1.2. Интерактивный алгоритм определения трансляции

Для отработки алгоритмов в качестве основных данных использовались временные серии бинарных файлов, содержащие информацию о горизонтальном поле радиолокационной отражаемости Z размером $400 \text{ км} \times 400 \text{ км}$ и пространственным шагом 4 км. Архивы таких серий подробно описаны нами в работе [16], где они использовались для анализа эволюции 264 МКС. Эти же дополненные архивы использовались для интерактивного определения трансляции и разработки ряда динамических характеристик МКС [14, 15].

В качестве иллюстрации процедуры выбран один день 12 августа 1991 г., с характерными условиями, осложняющими определение движения: малыми скоростями движения, входом и выходом штормов в область обзора и др. Перед началом интерактивной процедуры, выбранные из архива поля Z визуализировались в цветовой гамме, удобной для восприятия интенсивности конвективных явлений, менявшейся от ярко-зеленого к желтому, от желтого к оранжевому и красному (рис. 1). Так, если ярко-зеленый цвет пикселя соответствует $Z=10 \text{ dBZ}$ или слоистообразным осадкам, то пиксели насыщенного желтого цвета с переходом на оранжевый соответствуют $Z = 40 \text{ dBZ}$ или конвективным осадкам грозовых штормов, оранжевые тона сигнализируют об отражаемостях Z , переходных от грозовых штормов к градовым $>50 \text{ dBZ}$, которые, в свою очередь, выделены оттенками красного. Опыт работы с изображениями показал [15], что наилучших результатов интерактивной процедуры можно добиться при «размножении истинных пикселей» с заменой одного пикселя $4 \text{ км} \times 4 \text{ км}$ на 4 пикселя $1 \text{ км} \times 1 \text{ км}$ той же отражаемости. В этом случае на программном окне 600×800 пикселей одновременно уместятся два последовательных во времени изображения поля осадков, и остается место командным строкам и вспомогательной информации. В дальнейшем предполагается, что интервал времени между изображениями, используемыми для определения трансляции, составляет около 30 мин., и оба изображения одновременно визуализированы на экране (рис. 1а). На рис. 1а отображен выбор сохраняющихся фрагментов поля относительно «мишени» – большого круга со штормами, сопровождающимися опасными явлениями.

Интерактивная структурная процедура, характеризуемая как «идентификация и сопровождение сохраняющихся мезо-β-масштабных фрагментов поля» заключается в следующих шагах. На первом шаге оператор подводит курсор, в нашем случае это прямоугольник 15×15 км (рис. 1а, справа), к центру интересующей области на первом изображении. В случае определения трансляции эта область представляет собой фрагмент поля, содержащий группу штормов на первом и втором изображении, до определенной степени сохраняющих взаимную конфигурацию. Наведением курсора на область интереса проще управлять с клавиатуры дисплея. Заметим, что на этом этапе, занимающем около 5 с, от оператора не требуется особых навыков и точности. Впрочем, чтобы оператор быстро ориентировался в выборе области, автоматически возникают «мишени» – области развития доминирующего на данный момент шторма с $Z > 50$ dBZ (большой круг на рис. 1а).

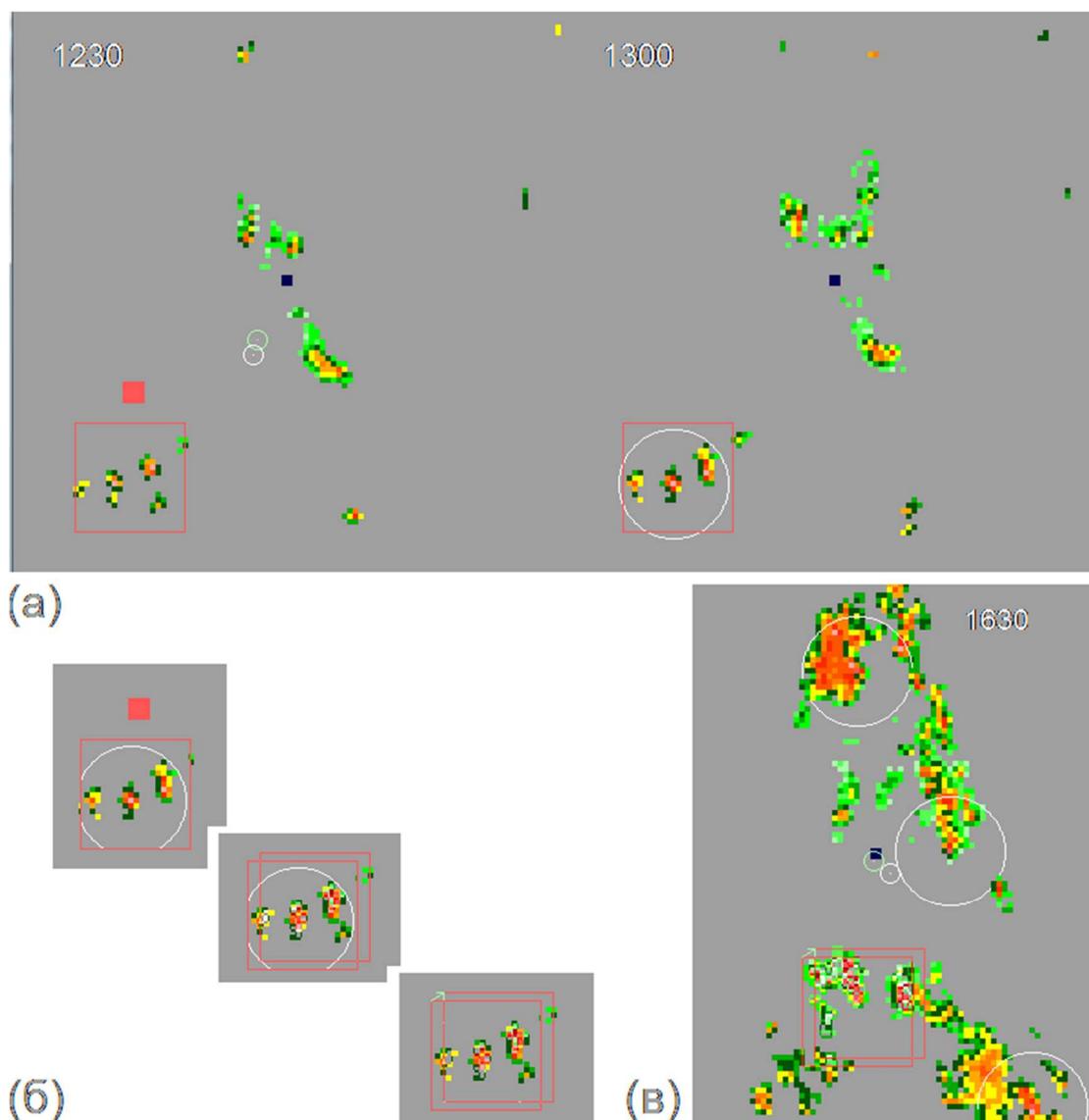


Рис. 1. Интерактивная процедура определения трансляции (ПОТ):
 а) Выбор сохраняющихся фрагментов поля относительно «мишени»;
 б) отдельные шаги ПОТ; в) ПОТ в 16:30-17:00

Как только центр области интереса выбран, начинается заключительный этап процедуры. При этом на обоих изображениях (рис. 1а) появляются окна достаточного размера, чтобы в них попали несколько штормов из области интереса. Размер окна определяется максимальными масштабами доминирующей группы штормов ≈ 80 км [5, 11, 12] и максимальной скоростью трансляции ≈ 100 км/ч. Так, при интервале между изображениями ≈ 30 минут практически в любых погодных ситуациях можно использовать квадратное окно 80 км \times 80 км. Информация о поле Z в окнах автоматически заносится в память как сегмент изображения, а затем сегмент более позднего по времени изображения поля Z появляется поверх более раннего слоя, как отдельный полупрозрачный слой (рис. 1б). На рис. 1б показаны отдельные шаги ПОТ: появление фрагментов, их совмещение и расчета вектора трансляции (внизу). Оператору остается «наилучшим» образом совместить последний (верхний) и прошлый (нижний) фрагменты поля, сдвигая верхний фрагмент относительно нижнего по горизонтали и вертикали. При вычислении трансляции задача наилучшего совмещения заключается в наложении фрагментов одного и того же эволюционирующего поля таким образом, чтобы совпали осадки штормов присутствующие на обоих снимках.

В зависимости от опытности оператора на попытки совмещения уходит от 5 до 30 с, по завершении которых автоматически выдается скорость и направление смещения фрагмента (обозначено стрелкой на рис. 1б). Это и есть первичная оценка вектора трансляции. При необходимости¹, оценку можно повторить, выбрав новые фрагменты поля и получив, в конце концов, некоторую среднюю оценку для данного момента времени. В общем случае рекомендуется повторять эту процедуру по мере поступления новой радарной информации, но не реже 1 раза в час, рекурсивно фильтруя «выбросы» оценок. Обычно достаточно трехточечной сглаживающей фильтрации.

Кажущаяся субъективность процесса совмещения сохраняющихся фрагментов компенсируется стабильной оценкой движения практически для любых систем осадков. Действительно, таким способом удается получить оценку движения системы во всех случаях, когда автоматические процедуры несостоятельны. Нетрудно показать, что природа «отказов процедур» обусловлена многомодальной эволюцией элементов поля осадков, своеобразным «цветным шумом». Это, в частности, иллюстрируется на рис. 1а и 1в, где приведены автоматические оценки положения центра масс поля МКС и рассчитано его движение. ПОТ в 16:30–17:00 демонстрируют ту же скорость трансляции, что и на рис. 1а и рис. 1б, но движение центра МКС совершенно другое. Так, в 1230–1300 со скоростью 23 км/ч центр поля сдвинулся из азимута 195° , а штормы транслировались со скоростью 22 км/ч на северо-восток из азимута 236° . Однако, в 1630–1700, из-за развития штормов на северо-западе и вхождения в обзор радара с юго-востока новых доминант, центр масс поля сдвигался на северо-запад из азимута 144° со скоростью 31 км/ч, при примерно том же направлении трансляции.

1.3. Объектно-ориентированные автоматические процедуры

Учитывая ограничения автоматических структурных процедур на сроках, превышающих 15 мин, нами исследованы возможности идентификации и определения движения на

¹ Выбор новых фрагментов на одной и той же паре снимков целесообразно проводить для МКС с явно выраженным центром вращения: ураганы и им подобные внетропические системы низкого давления.

основе стандартных алгоритмов кластеризации (сегментации), таких как k -means [20] и k -medoids [21]. Особенность этих алгоритмов – это выделение объектов, зависящее только от задаваемого числа кластеров k и выбранной для их разделения метрики (обычно это физическое расстояние). Поэтому число элементов k можно задать с учетом иерархии конвективных систем. Так, в α -кластере размером ≈ 300 км за период конвективной активности 6-9 часов [5, 11, 12] возникает $\approx 2-3$ доминантных ансамбля ≈ 100 км, или в три раза больше доминантных штормов ≈ 25 км. Таким образом, для сегментации изображения одной МКС, в зависимости от детальности необходимо задать от 4 до 10 сегментов меньшего размера.

Для ускорения в качестве начальных координат поиска можно задать k -мощных штормов, удаленных на расстояние более 20 км. После окончания итераций координаты центров кластеров сохраняются, а цикл кластеризации повторяется для нового изображения. Вопрос об идентификации одних и тех же кластеров на соседних изображениях решен минимизацией расстояний между старыми и новыми положениями центров кластеров. Разработанное программное обеспечение позволяло одновременно проводить расчеты скоростей кластеров, определенных двумя методами [19, 20] и по описанной выше интерактивной процедуре.

Общий результат этих экспериментов иллюстрируется следующим образом. При 2-3-кратном повторении интерактивной процедуры в исследуемый день по 30 получасовым интервалам получена средняя скорость трансляции $22,4 \pm 4,9$ км/ч. Предполагая, что оценка трансляции может быть получена как средняя скорость поля или кластеров, в тех же условиях получено, что средняя скорость центра масс поля была равна $32,2 \pm 24,6$ км/ч, средняя скорость четырех медоид составляла $19,6 \pm 7,4$ км/ч, а средняя скорость четырех кластеров k -means $19,4 \pm 12,9$ км/ч. В целом это означает большую устойчивость оценки скорости по интерактивной процедуре и перспективность кластеризации при условии, если будет найден способ, фильтрующий сильные скачки в положении отдельных кластеров.

2. Сверхкраткосрочный прогноз и его верификация

Консервативность трансляции – залог ее использования в прогнозе опасных явлений погоды на сроки до одного часа. В этом разделе описываются элементы экстраполяционного прогноза (nowcasting) и специально созданного для верификации прогноза композиционного изображения «жизненного цикла», построенного на основе прогнозируемого и реального поля осадков. Следует заметить, что идея такого рода композиционных изображений восходит к работе [4] и затем в различных вариантах широко использовалась во многих работах [5, 8, 11-13, 15]. Новизной представляемого варианта [15] является подчеркнутая его самодостаточность как метода оценки качества прогноза и как метода предварительной обработки поля для последующей экстраполяции.

2.1. Процедура «nowcasting»

Как первое приближение транслирование текущего поля осадков в новое положение, используется во всех современных системах штормового оповещения [1-3]. Так, последнее поле Z в 13:00 (рис. 1а), транслировано в прогнозируемое положение F за 14:00 (рис. 2а). Затем на основе статистики доминирующих штормов [12, 15] в поле F выделены две предполагаемые зоны опасных явлений протяженностью ≈ 30 км.

Изображение F на рис. 2а получено автоматически одночасовой трансляцией поля Z за 13 часов со скоростью 22 км/ч из азимута 236°. Перекрестие – это «новое» положение радара в 14:00, правильные восьмиугольники – области опасных явлений. R – реально наблюдаемое поле Z в 14:00, а LCD – композиционная обработка изображений F и R. Интерпретация зон осадков, отмеченных темно-красным, серым и цветовой палитрой на LCD дана в тексте. На рис. 2б то же, что и на рис. 2а, но в 18:00, на стадии зрелости МКС.

Обсуждение метода универсального прогноза этих явлений составляет предмет [12] и выходит за рамки данной работы. Здесь лишь отметим, что положение одного из октагонов, обусловлено типичным развитием штормов, а положение второго учитывает развитие ансамбля в целом. Предполагается, что эти правильные восьмиугольники будут информировать о вероятности восьми категорий опасных явлений, таких как град, крупный град, порывы, сильные шквалы, суммы осадков и т.п. Вероятностный прогноз этих явлений может быть получен по традиционной весовой методике применения функций нечеткой логики в прогнозах [2, 12, 22] или же, как предложено в работе [14], по степени принадлежности к климатологическим распределениям динамических характеристик систем.

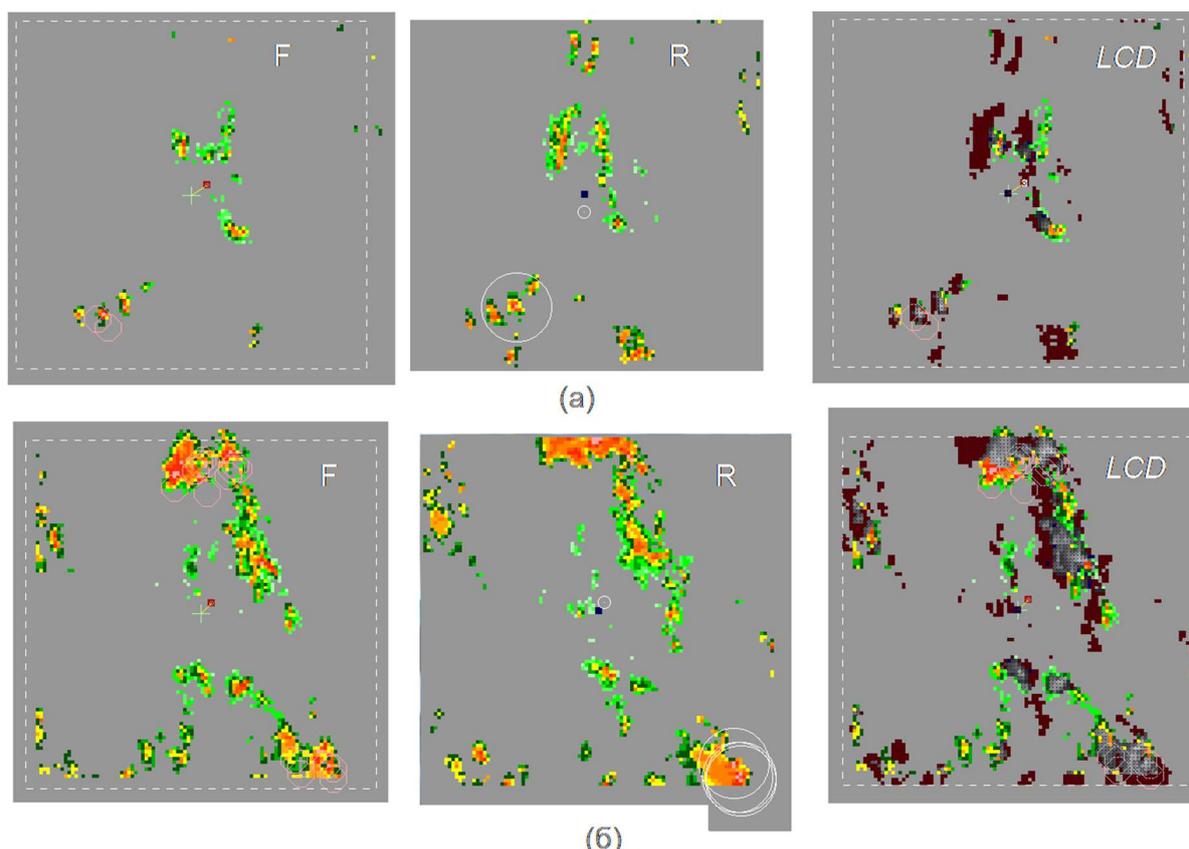


Рис. 2. Сверхкраткосрочный прогноз поля осадков и «дисплей жизненного цикла» (LCD), адаптированный для оценки прогноза с использованием реального изображения

Сопоставление прогнозной области опасных явлений F и реального R полей за 14:00, показывает, что на данный срок положение доминирующих штормов было предсказано относительно неплохо. Этот «удовлетворительный» результат получен вопреки тому, что жизненный цикл индивидуальных штормов составляет не более 1,5 часов. Дело в том, что ансамбли локальных штормов, имеющие масштабы ≈ 100 км, обычно доминируют 3–4 часа

[5, 8, 11, 12, 14], поэтому вполне оправданно предсказание того, что новый мощный шторм появится рядом с предыдущим. С другой стороны, анализируя реальное изображение R за 14:00, можно заметить новый ансамбль штормов в 150–200 км к северу от радара. Очевидно, что простой экстраполяцией появление этих штормов предсказать нельзя. Однако, именно этот северный ансамбль и второй, входящий в зону обзора радара с юга, будет доминировать на протяжении следующих 4 часов, сопровождаясь опасными явлениями (рис. 2б).

2.2. Дисплей жизненного цикла

Оценка качества прогноза мезомасштабных осадков – это одна из наиболее сложных методических проблем [14, 23] обусловленных тем, что осадки должны рассматриваться одновременно и как непрерывный процесс, и как локальное явление. Как показано в [15], оценку качества экстраполяционного прогноза удобнее проводить с помощью дисплея жизненного цикла (Life-Cycle Display, LCD) – особого композиционного изображения, имеющего три информативных слоя. Первый слой – это поле осадков прогностического изображения F . На слой F накладывается второй слой, который является производной от реально наблюдаемого поля R . Наложение происходит таким образом, что зоны осадков отсутствующие в прогнозируемом поле F , но присутствующие в поле R маркируются определенным цветом, например темно-красным как на рисунке 2. Третий слой – это области осадков, присутствующие в обоих слоях (выделены шкалой серости).

Детальное описание использования LCD в прогнозе выходит за рамки данной работы, поэтому мы отметим только главные моменты созданной композиции с точки зрения верификации прогноза, «цветные» пиксели LCD – это площади с переоценкой осадков, т.е. там, где дожди прогнозировались, но в срок прогноза осадков нет; темно-красные пиксели – это площади с недооценкой осадков, т.е. там, где дожди идут, но не прогнозировались, и, наконец, серые – это области удачного прогноза. Эти площади (число пикселей) получаются автоматически и соответствуют элементам матрицы верификации прогноза осадков, представленной в таблице.

Таблица

Матрица верификации прогноза осадков и ее соответствие дисплею жизненного цикла

	Осадки наблюдались	Осадки не наблюдались
Осадки прогнозировались (Hit+ False alarm)	Осуществившийся прогноз осадков равен числу серых пикселей (Hit)	Число неоправдавшихся прогнозов явления равно числу цветных пикселей (False alarm)
Осадки не прогнозировались (Miss+Null)	Число не оправдавшихся прогнозов отсутствия осадков равно числу темно-красных пикселей (Miss)	Число мест, где осадков не было и они не прогнозировались, равно числу черных пикселей (Null)

Применяя таблицу и вычисляя число пикселей того или иного цвета LCD, можно рассчитать три следующие традиционные меры качества прогноза.

- 1) *Отношение ложной тревоги (false of alarm ratio, FAR)* – это отношение числа не оправдавшихся прогнозов ко всей сумме прогнозов явлений. Сумма прогнозов – это

число пикселей, в которых прогнозировались осадки, т.е. осадки поля F ; на изображении LCD – это сумма серых и цветных пикселей. Неоправдавшиеся прогнозы явления – это случаи, когда осадки были спрогнозированы, однако в срок не наблюдались, т.е. цветные пиксели: $FAR = \frac{False\ alarm}{Hit + False\ alarm}$.

2) *Вероятность обнаружения (probability of detection, POD)* – это другая мера точности прогноза, учитывающая число успешно спрогнозированных (hit) пикселей ко всему числу пикселей с осадками, наблюдавшимися в срок, т.е. к полю R . На LCD мера POD будет выражаться как число серых пикселей (hit) по отношению к сумме серых (hit) и темно-красных (miss): $POD = \frac{Hit}{Hit + Miss}$.

3) *Критический индекс успешности (critical success index, CSI)* комбинирует FAR и POD в одну статистику: $CSI = \frac{Hit}{Hit + Miss + False\ alarm}$. CSI – это число «серых пикселей» к общему числу пикселей с наблюдаемыми и прогнозируемыми осадками на LCD.

Таким образом, LCD является простым и эффективным способом визуализации качества прогноза. Например, сравнивая LCD рис. 2, можно увидеть, что относительные площади успешного прогноза заметно выше в более позднюю стадию МКС.

Интерпретацию рис. 2, можно провести в прогностическом аспекте, представляя цвет пикселя на LCD как стадию жизни конвективных штормов. Поскольку на LCD, темно-красным цветом отмечены конвективные осадки, появившиеся в течение последнего времени, а серым – дождь переходный с ливневого на умеренный морозящий, то ориентируясь на взаимное положение этих областей, можно оценить направление развития новых зон глубокой конвекции, а также их диссипации. Например, развитие новых кучево-дождевых облаков невозможно в транслируемых областях, где недавно прошли осадки [24]. Отсюда следует, что экстраполяционная процедура может быть существенно скорректирована, и вместо трансляции всего поля можно переносить только зоны потенциального развития элементов МКС.

Заключение

В работе представлены элементы альтернативной системы оповещения об опасных явлениях погоды, основанной на обработке радиолокационных данных и состоящей из трех программных процедур: определения скорости трансляции, экстраполяционного прогноза и верифицирующего его качество дисплея жизненного цикла. Основные преимущества этой системы заключается в том, что:

- (i) простая полуавтоматическая процедура, использующая смещение консервативных фрагментов поля, дает устойчивые оценки трансляции для всей мезомасштабной системы;
- (ii) в свою очередь консервативность оценок трансляции позволяет экстраполировать текущее поле осадков на сроки до 1 часа и адекватно интерпретировать композиционные изображения дисплея жизненного цикла LCD;
- (iii) как в терминах оценок качества прогноза;
- (iv) так и отразить области развития и диссипации штормов, а так же зоны запрета на развитие новых очагов конвекции.

Заметим, что разрабатываемая система оповещения об опасных явлениях конструктивно совместима с разработанными ранее методами анализа МКС [26-32]. В том числе система может быть дополнена методом анализа данных доплеровского радара Meap

Wind Relative, когда компонента трансляции вычитается из поля радиальных скоростей [25-29], методом выделения стадий МКС по данным геостационарных спутников [30] и комбинирована анализом данных систем регистрации гроз [31] и другими способами.

Следует надеяться, что продолжение данных работ позволит найти: а) более надежные автоматические алгоритмы определения трансляции; б) на основе информации LCD построить новые функции принадлежности прогноза и в) создать простейшую систему ассимиляции радиолокационных данных в численную модель прогноза WRF-ARW, недавно освоенную в Лаборатории суперкомпьютерного моделирования ЮУрГУ [14, 32].

Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации в рамках соглашения № 14.В37.21.0613.

Литература

1. Clifford, M. Nowcasting: The Promise of New Technologies of Communication, Modeling, and Observation / M. Clifford // Bulletin of the American Meteorological Society. — 2012. — Vol. 93, No. 6. — P. 797–809.
2. Mueller, C. NCAR Autonowcast System / C. Mueller, T. Saxen, R. Roberts, et al. // Weather and Forecasting. — 2003. — Vol. 18, No. 4. — P. 545–561.
3. Lakshmanan, V. The Warning Decision Support System-Integrated Information / V. Lakshmanan, T. Smith, G. Stumpf, et al. // Weather and Forecasting. — 2007. — Vol. 22. — P. 596–612.
4. Старостин, А.Н. Мезомасштабная структура полей радиоэха конвективных облаков в Молдавии / А.Н. Старостин, Е.М. Лившиц, В.С. Швецов // Метеорология и гидрология. — 1983. — № 10. — С. 55–59.
5. Abdoulaev, S. Evolution and Hierarchy of Cumulonimbus Ensembles / S. Abdoulaev // Brazilian Journal of Meteorology. — 1995. — Vol. 10. — P. 1–9.
6. Corfidi, S.F. Predicting of Movement of Mesoscale Convective Complexes / S.F. Corfidi, J.H. Merrit, J.M. Fritsch // Weather and Forecasting. — 1996. — Vol. 11. — P. 42–46.
7. Абдуллаев, С.М. Эволюционная классификация мезомасштабных линий шквала / С.М. Абдуллаев, О.Ю. Ленская // Метеорология и гидрология. — 1998. — № 3. — С. 24–32.
8. Abdoulaev, S. Mesoscale Precipitation Systems in Rio Grande do Sul. Part 3: Structure and Evolution of Non-Line Mesoconvective Systems / S. Abdoulaev, A. Starostin, O. Lenskaia // Brazilian Journal of Meteorology. — 2001. — Vol. 16, No. 2. — P. 87–102.
9. Corfidi, S.F. Cold Pools and MCS Propagation: Forecasting of Motion of Downwind – Developing MCSs / S.F. Cordifi // Weather and Forecasting. — 2003. — Vol. 18. — P. 992–1016.
10. Ленская, О.Ю. Методические вопросы использования спутниковой и радиолокационной информации в мезомасштабном прогнозе (на примере опасных явлений погоды в Москве 24 июля 2001 г.) / О.Ю. Ленская // Вестник Челябинского государственного университета. — 2007. — №6(84). Вып. 2. — С. 66–80.
11. Абдуллаев, С.М. Жизненный цикл мезомасштабных конвективных систем / С.М. Абдуллаев, А.А. Желнин, О.Ю. Ленская // Метеорология и гидрология. — 2009. — № 5. — С. 34–45.

12. Абдуллаев, С.М. Жизненный цикл мезомасштабных конвективных систем: концепция, климатология и прогноз: дис. ... докт. геогр. наук : 25.00.30 / С.М. Абдуллаев. — М.: Гидрометеорологический научно-исследовательский центр Российской Федерации, 2010. — 407 с.
13. Плюснин, С.Д. Прогноз эволюции конвективных штормов на основе анализа их радиозахв и годографа ветра в тропосфере в подвижной системе координат / С.Д. Плюснин, Е.И. Потапов, И.А. Гараба и др. // Метеорология и гидрология. — 2013. — № 7. — С. 43–52.
14. Lenskaya, O.Yu. Organization and Evolution of Mesoscale Convective Systems Using Radar Data: Objective Description. The «Dominating Thunderstorm» Conception and its Application to MCS Climatology / O.Yu. Lenskaya, S.M. Abdullaev, A.A. Zhelnin // Prepr. of 15th Conf. on Mesoscale Proc. (Portland, Oregon, 2013). URL: <https://ams.confex.com/ams/15MESO/webprogram/Paper227607.html> (дата обращения: 15.10.2013).
15. Abdullaev, S.M. A Diagnostic and Forecasting Techniques Based on Radar Derived Translation and Propagation of Convective Systems / S.M. Abdullaev, O.Yu. Lenskaia // Prepr. of 36th Conf. on Radar Meteorology: Amer. Meteor. Soc. (Breckenridge, CO, 2013). URL: <https://ams.confex.com/ams/36Radar/webprogram/Paper228139.html>.
16. Abdullaev, S.M. The Structure of Mesoscale Convective Systems in Central Russia / S.M. Abdullaev, O.Y. Lenskaya, A.A. Zhelnin // Russian Meteorology and Hydrology. — 2012. — Vol. 37. No. 1. — P. 12–20.
17. Han, L. 3D Convective Storm Identification, Tracking and Forecasting – An Enhanced TITAN Algorithm / L. Han, S. Fu, L. et al. // Oceanic Technol. — 2009. — Vol. 26. — P. 719–732.
18. Lakshmanan, V. An Objective Method of Evaluating and Sevising Storm-Tracking Algorithms / V. Lakshmanan, T. Smith // Weather and Forecasting. — 2010. — Vol. 25. — P. 701–709.
19. Республиканский гидрометеоцентр. URL: http://pogoda.by/thumbnails/?map=radar_map (дата обращения: 15.10.2013).
20. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. / J. MacQueen // Proc. Fifth Berkeley Sympos. Math. Statist. and Probability (Berkeley, Calif., 1965/66). — Univ. California Press, Berkeley, Calif., 1967. — Vol. I: Statistics. — P. 281–297.
21. Kaufman, L. Clustering by Means of Medoids. In Statistical Data Analysis Based on the L1-Norm and Related Methods. / L. Kaufman, P.J. Rousseeuw, edited by Y. Dodge. — 1987. — P. 405–416.
22. Гаязова, А.О. Прогнозирование численности *Microcystis Aeruginosa* на основе правил нечеткой логики и нечетких нейронных сетей / А.О. Гаязова, С.М. Абдуллаев // Вестник Южно-Уральского государственного университета. — 2012. — Т. 47. — С. 5–11.
23. Lakshmanan, V. A Gaussian Mixture Model Approach to Forecast Verification / V. Lakshmanan, J.S. Kain // Weather and Forecasting. — 2010. — Vol. 25, No. 3. — P. 908–920.

24. Starostin, A. Forecast of Storm Dissipation / A. Starostin, S. Abdoulaev // Prepr. of 7th Conf. on Mesoscale Proc. (United Kingdom, 9–13 September). — UK: Reading UK, 1996. — P. 399–400.
25. Abdoulaev, S. Mean Wind Relative Motions and Typical Evolution of Mesoscale Convective Systems Having Complex Organization / S. Abdoulaev, O. Lenskaia, A. Zhelnin, // Prepr. of 8th Conf. on Mesoscale Proc. (Boulder, Colorado, 1999). — P. 115–116.
26. Abdoulaev, S. Structure of Motions in Linear Mesoscale Convective Systems Accompanied by Stratiform Region / S. Abdoulaev, O. Lenskaia // Prepr. 8th Conf. on Mesoscale Proc. (Boulder, Colorado, 1999). — P. 113–114.
27. Abdoulaev, S. Relative Motions in Squall Lines Accompanied by Stratiform Region / S. Abdoulaev, O. Lenskaia, V.S. Marques, et al. // Brazilian Journal of Meteorology. — 2000. — Vol. 15, No. 2. — P. 87–102.
28. Abdoulaev, S. Doppler Radar Study of Quasi-Stationary Mesoscale Frontal Systems. Part 1: Periodical structures / S. Abdoulaev, O. Lenskaia, V.S. Marques, et al. // Brazilian Journal of Meteorology. — 2002. — Vol. 17, No. 1. — P. 53–68.
29. Abdoulaev, S. Doppler Radar Study of Quasi-Stationary Mesoscale Frontal Systems. Part 2: Transversal movements / S. Abdoulaev, O. Lenskaia, V.S. Marques, et al. // Brazilian Journal of Meteorology. — 2002. — Vol. 17, No. 1. — P. 69–82.
30. Lenskaia, O. Mesoscale Precipitation Systems in Rio Grande do Sul. Part 2: Synoptic and Satellite Overview / O. Lenskaia, J. Marques, S. Abdoulaev // Prepr. of 28th Conference on Radar Meteorology (Austin, USA, 7–12 September 1997). — P. 489–490, 592–593.
31. Abdoulaev, S. Analysis of Mesoscale System Using Cloud-to-ground Flash Data / S. Abdoulaev, V.S. Marques, M.A. Pinheiro, et al. // Brazilian Journal of Geophysics. — 2001. — Vol. 19, No. 1. — P. 75–95.
32. Ленская, О.Ю. Численное моделирование характеристик пограничного слоя атмосферы крупного промышленного города (на примере г. Челябинска) / О.Ю. Ленская, С.М. Абдуллаев, А.И. Приказчиков и др. // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». — 2013. — Том 2. — № 2. — С. 65–82.

Абдуллаев Санжар Муталович, доктор географических наук, профессор кафедры вычислительной математики, Южно-Уральский государственный университет (Челябинск, Российская Федерация), sanjar@mail.ru

Ленская Ольга Юрьевна, кандидат географических наук, доцент, доцент кафедры вычислительной математики, Южно-Уральский государственный университет (Челябинск, Российская Федерация), lenskaia_olga@mail.ru

Гаязова Анна Олеговна, магистр экологии, аспирант кафедры вычислительной математики, Южно-Уральский государственный университет (Челябинск, Российская Федерация), why.ann@mail.ru

Соболев Дмитрий Николаевич, магистр вычислительной математики и информатики, кафедра вычислительной математики, Южно-Уральский государственный университет (Челябинск, Российская Федерация), propp.ds@gmail.com

Носков Артем Анатольевич, бакалавр информационных технологий, кафедра вычислительной математики, Южно-Уральский государственный университет (Челябинск, Российская Федерация), artem.noskov@gmail.com

Иванова Ольга Николаевна, кандидат педагогических наук, доцент кафедры системного программирования, Южно-Уральский государственный университет (Челябинск, Российская Федерация), ivanovaon@list.ru

Радченко Глеб Игоревич, кандидат физико-математических наук, и.о. декана факультета вычислительной математики и информатики, Южно-Уральский государственный университет (Челябинск, Российская Федерация), gleb.radchenko@gmail.com

Поступила в редакцию 15 октября 2013 г.

*Bulletin of the South Ural State University
Series "Computational Mathematics and Software Engineering"
2014, vol. 3, no. 1, pp. 17–32*

SHORT-RANGE FORECASTING ALGORITHMS USING RADAR DATA: TRANSLATION ESTIMATE AND LIFE-CYCLE COMPOSITE DISPLAY

S.M. Abdoulaev, South Ural State University (Chelyabinsk, Russian Federation)

O.U. Lenskaia, South Ural State University (Chelyabinsk, Russian Federation)

A.O. Gayazova, South Ural State University (Chelyabinsk, Russian Federation)

O.N. Ivanova, South Ural State University (Chelyabinsk, Russian Federation)

A.A. Noskov, South Ural State University (Chelyabinsk, Russian Federation)

D.N. Sobolev, South Ural State University (Chelyabinsk, Russian Federation)

G.I. Radchenko, South Ural State University (Chelyabinsk, Russian Federation)

The mesoscale convective system (MCS) cinematic and evolution routinely derived from real-time radar reflectivity Z data are used to develop 3 step short-range forecasting technique. The first forecasting step is estimation of MCS translation velocity. It is proposed that translation can be estimated by robust interactive procedure, identifying and tracking of "conservative meso- β fragments", or by automatic clustering algorithms. The second step is nowcasting when entire Z field is extrapolated by translation to forecast time $\sim 0,5-1$ hours. The third step is forecast verification and MCS diagnostic when prognostic and real-time images of Z are coupled to one composite image, denominated as life-cycle display (LCD). The construction of LCD is adapted to use both in forecast verification in terms of FAR, POD and CSI and to display 3 type conventionally outlined MCS areas where: 1) ongoing convection occur; 2) dissipating convection/stratiform transformation and 3) precipitation was recently ceased. As poor extrapolation forecasts associated, mainly, with growth and decay of storms in the forecast period, life-cycle composite images provide valuable information about of developing, dissipating areas and areas prohibited to new convective development.

Keywords: mesoscale convective system, radar precipitation, motion algorithm, severe weather nowcasting, forecast verification, life-cycle display

References

1. Clifford M. Nowcasting: The Promise of New Technologies of Communication, Modeling, and Observation // *Bulletin of the American Meteorological Society*. 2012. Vol. 93, No. 6. P. 797–809.
2. Mueller C., Saxen T., Roberts R., Wilson J., Betancourt T., Dettling S., Quin N., Yee J. NCAR Autowcast System // *Weather and Forecasting*. 2003. Vol. 18, No. 4. P. 545–561.
3. Lakshmanan V., Smith T., Stumpf G., Hondl K. The Warning Decision Support System-Integrated Information // *Weather and Forecasting*. 2007. Vol. 22. P. 596–612.
4. Starostin A.N., Livshic E.M., Shvecov V.S. Mezomasshtabnaja struktura polej radiojeha konvektivnyh oblakov v Moldavii [Mesoscale Structure of the Fields of Radioecho of Convective Clouds in Moldova]. *Meteorologija i gidrologija [Meteorology and Hydrology]*. 1983. No. 10. P. 55–59.
5. Abdoulaev S. Evolution and Hierarchy of Cumulonimbus Ensembles // *Brazilian Journal of Meteorology*. 1995. Vol. 10. P. 1–9.
6. Corfidi S.F., Merrit J.H., Fritsch J.M. Predicting of Movement of Mesoscale Convective Complexes // *Weather and Forecasting*. 1996. Vol. 11. P. 42–46.
7. Abdullaev S.M., Lenskaja O.Yu. Jevoljucionnaja klassifikacija mezomasshtabnyh linij shkvala [Evolutionary Classification of Mesoscale Squall Lines]. *Meteorologija i gidrologija [Meteorology and Hydrology]*. 1998. No. 3. P. 24–32.
8. Abdoulaev S., Starostin A., Lenskaia O. Mesoscale Precipitation Systems in Rio Grande do Sul. Part 3: Structure and evolution of non-line mesoconvective systems // *Brazilian Journal of Meteorology*. 2001. Vol. 16, No. 2. P. 87–102.
9. Corfidi S.F. Cold Pools and MCS Propagation: Forecasting of Motion of Downwind – Developing MCSs // *Weather and Forecasting*. 2003. Vol. 18. P. 992–1016.
10. Lenskaja O.Yu. Ispol'zovanie doplerovskogo radara i dannyh nazemnyh nabljudenij dlja izuchenija i prognoza shkvalov [The Use of Doppler Radar and Surface Observations for Studying and Forecasting the Squalls]. *Vestnik Chelyabinskogo gosudarstvennogo universiteta [Bulletin of Chelyabinsk State University]*. 2005. No. 1(1). Vol. 1. P. 131–143.
11. Lenskaja O.Yu. Metodicheskie voprosy ispol'zovanija sputnikovoj i radiolokacionnoj informacii v mezomasshtabnom prognoze (na primere opasnyh javlenij pogody v Moskve 24 ijulja 2001 g.) [Methodological Issues of the Use of Satellite and Radar Data in Mesoscale Forecast (by the Example of Severe Weather Events in Moscow on July 24, 2001)]. *Vestnik Chelyabinskogo gosudarstvennogo universiteta [Bulletin of Chelyabinsk State University]*. 2007. No. 6(84). Vol. 2. P. 66–80.
12. Abdullaev S.M., Zhelnin A.A., Lenskaja O.Yu. Zhiznennyj cikl mezomasshtabnyh konvektivnyh system [The Life Cycle of Mesoscale Convective Systems]. *Meteorologija i gidrologija [Meteorology and Hydrology]*. 2009. No. 5. P. 34–45.
13. Abdullaev S.M. Zhiznennyj cikl mezomasshtabnyh konvektivnyh sistem: koncepcija, klimatologija i prognoz [The Life Cycle of Mesoscale Convective Systems: Concept, Climatology and Forecast]. Moscow, 2010. 407 P.
14. Pljusnin S.D., Potapov E.I., Garaba I.A., Popova V.P. Prognoz jevoljucii konvektivnyh shtormov na osnove analiza ih radiojeha i godografa vetra v troposfere v podvizhnoj sisteme koordinat [Forecast Evolution of Convective Storms on the Basis of their Radio Echo

- and Hodograph of the Wind in the Troposphere in the Moving Coordinate System]. *Meteorologija i gidrologija* [Meteorology and Hydrology]. 2013. No. 7. P. 43–52.
15. Lenskaya O.Yu., Abdullaev S.M., Zhelnin A.A. Organization and Evolution of Mesoscale Convective Systems Using Radar Data: Objective Description. The «Dominating Thunderstorm» Conception and its Application to MCS Climatology // Prepr. of 15th Conf. on Mesoscale Proc. (Portland, Oregon, 2013). URL: <https://ams.confex.com/ams/15MESO/webprogram/Paper227607.html> (accessed: 15.10.2013).
 16. Abdullaev S.M., Lenskaia O.Yu. A Diagnostic and Forecasting Techniques Based on Radar Derived Translation and Propagation of Convective Systems // Prepr. of 36th Conf. on Radar Meteorology: Amer. Meteor. Soc. (Breckenridge, CO, 2013). URL: <https://ams.confex.com/ams/36Radar/webprogram/Paper228139.html> (accessed: 15.10.2013).
 17. Abdullaev S.M., Lenskaya O.Y., Zhelnin A.A. Struktura mezomasshtabnyh konvektivnyh sistem v central'noj Rossii [Structure of Mesoscale Convective Systems in Central Russia]. *Meteorologija i gidrologija* [Meteorology and Hydrology]. 2012. No. 1. P. 20–32.
 18. Han L., Fu S., Zhao L., Zheng Y., Wang H., Lin Y. 3D Convective Storm Identification, Tracking and Forecasting – An Enhanced TITAN Algorithm // *Oceanic Technol.* 2009. Vol. 26. P. 719–732.
 19. Lakshmanan V., Smith T. An Objective Method of Evaluating and Devising Storm-tracking Algorithms // *Weather and Forecasting.* 2010. Vol. 25. P. 701–709.
 20. Respublikanskij gidrometeocentr [Republican Hydrometeorological Center]. URL: http://pogoda.by/thumbnails/?map=radar_map (accessed: 15.10.2013).
 21. MacQueen J. Some methods for classification and analysis of multivariate observations. In: Proc. / Fifth Berkeley Sympos. Math. Statist. and Probability (Berkeley, Calif., 1965/66). Univ. California Press, Berkeley, Calif., 1967. Vol. I: Statistics. P. 281–297.
 22. Kaufman L., Rousseeuw P.J. Clustering by Means of Medoids. In *Statistical Data Analysis Based on the L1-Norm and Related Methods.* 1987. P. 405–416.
 23. Gajazova A.O., Abdullaev S.M. Prognozirovanie chislennosti *Microcystis Aeruginosa* na osnove pravil nechetkoj logiki i nechetkih nejronnyh setej [Forecasting of *Microcystis Species Aeruginosa* Rule-Based Fuzzy Logic and Fuzzy Neural Networks]. *Vestnik Yuzho-Uralskogo gosudarstvennogo universiteta. Seriya "Matematicheskoe modelirovanie i programmirovanie"* [Bulletin of South Ural State University. Series: Mathematical Modeling, Programming & Computer Software]. 2012. T. 47. P. 5–11.
 24. Lakshmanan V., Kain J.S. A Gaussian Mixture Model Approach to Forecast Verification // *Weather and Forecasting.* 2010. Vol. 25, No. 3. P. 908–920.
 25. Starostin A., Abdoulaev S. Forecast of Storm Dissipation // Prepr. of 7th Conf. on Mesoscale Proc. (United Kingdom, 9-13 September). UK: Reading UK, 1996. P. 399–400.
 26. Abdoulaev S., Lenskaia O., Zhelnin A. Mean Wind Relative Motions and Typical Evolution of Mesoscale Convective Systems Having Complex Organization // Prepr. of 8th Conf. on Mesoscale Proc. (Boulder, Colorado, 1999). P. 115–116.
 27. Abdoulaev S., Lenskaia O. Structure of Motions in Linear Mesoscale Convective Systems Accompanied by Stratiform Region // Prepr. 8th Conf. on Mesoscale Proc. (Boulder, Colorado, 1999). P. 113–114.

28. Abdoulaev S., Lenskaia O., Marques V.S., Pinheiro F.M.A. Relative Motions in Squall Lines Accompanied by Stratiform Region // Brazilian Journal of Meteorology. 2000. Vol. 15, No. 2. P. 87–102.
29. Abdoulaev S., Lenskaia O., Marques V.S., Pinheiro F.M.A. Doppler Radar Study of Quasi-Stationary Mesoscale Frontal Systems. Part 1: Periodical Structures // Brazilian Journal of Meteorology. 2002. Vol. 17, No. 1. P. 53–68.
30. Abdoulaev S., Lenskaia O., Marques V.S., Pinheiro F.M.A. Doppler Radar Study of Quasi-Stationary Mesoscale Frontal Systems. Part 2: Transversal Movements // Brazilian Journal of Meteorology. 2002. Vol. 17, No. 1. P. 69–82.
31. Lenskaia O., Marques J., Abdoulaev S. Mesoscale Precipitation Systems in Rio Grande do Sul. Part 2: Synoptic and Satellite Overview // Prepr. of 28th Conference on Radar Meteorology (Austin, USA, 7-12 September 1997). P. 489-490, 592–593.
32. Abdoulaev S., Marques V.S., Pinheiro M.A., Martinez E.F., Lenskaia O. Analysis of Mesoscale System Using Cloud-to-Ground Flash Data // Brazilian Journal of Geophysics. 2001. Vol. 19, No. 1. P. 75–95.
33. Lenskaja O.Yu. Chislennoe modelirovanie harakteristik pogranichnogo sloja atmosfery krupnogo promyshlennogo goroda (na primere g. Cheljabinska) [Numerical Modeling of the Characteristics of the Atmospheric Boundary Layer of a Large Industrial City (on the Example of Chelyabinsk)]. Vestnik Yuzho-Uralskogo gosudarstvennogo universiteta. Seriya «Matematicheskoe modelirovanie i programmirovaniye» [Bulletin of South Ural State University. Series: Mathematical Modeling, Programming & Computer Software]. 2013. T. 2. No. 2. P. 65–82.

Received 15 October 2013

МОДЕЛИРОВАНИЕ СЕЙСМИЧЕСКОГО ПОЛЯ В АКУСТИЧЕСКОМ ПРИБЛИЖЕНИИ ДВУХФАЗНЫХ, ИЕРАРХИЧЕСКИ НЕОДНОРОДНЫХ СРЕД

О.А. Хачай, А.Ю. Хачай

В работе рассмотрена задача моделирования сейсмического поля (случай распространения продольной волны с учетом только сжатия и растяжения среды) в слоистой среде с включениями иерархической структуры. Построены алгоритмы моделирования в сейсмическом случае для 2-D упругой и пористой иерархической неоднородности. Исследован вопрос отражения физических свойств однофазности и двухфазности в системе уравнений решения прямой динамической сейсмологии в частотном варианте.

Ключевые слова: иерархическая, упругая, пористая, двухфазная среда, сейсмическое поле, алгоритмы моделирования.

Введение

Процессы разработки нефтегазовых месторождений связаны с движением многофазных многокомпонентных сред, которые характеризуются неравновесными и нелинейными реологическими свойствами. Реальное поведение пластовых систем определяется сложностью реологии движущихся жидкостей и морфологического строения пористой среды, а также многообразием процессов взаимодействия между жидкостью и пористой средой [1]. Учет этих факторов необходим для содержательного описания процессов фильтрации за счет нелинейности, неравновесности и неоднородности, присущих реальным системам. При этом выявляются новые синергетические эффекты (потеря устойчивости с возникновением колебаний, образование упорядоченных структур). Это позволяет предложить новые методы контроля и управления сложными природными системами, которые настроены на учет этих явлений. Таким образом, пластовая система, из которой необходимо извлечь нефть, представляет собой сложную динамическую иерархическую систему.

При построении математической модели реального объекта необходимо в качестве априорной информации использовать данные активного и пассивного мониторинга, получаемые в ходе текущей эксплуатации объекта. Решение обратных задач имеет огромное значение для нефтяной промышленности, поскольку нефтяной пласт относится к числу природных систем, не поддающихся прямым измерениям и наблюдениям в целом. Исследования последних лет показали, что в эволюции динамических систем играют неустойчивости, природу которых изучает теория самоорганизации или синергетика. Информацию об их проявлениях в нефтяном пласте при его отработке можно только получить, используя данные мониторинга, чувствительные к его иерархической структуре.

Настоящая работа посвящена выводу интегральных уравнений двумерной прямой задачи для сейсмического поля в динамическом варианте. В работе произведен совместному анализу интегральных уравнений двумерных задач для сейсмического поля в рамках модели локальной иерархической неоднородности с пористым включением и чисто упругой неоднородности иерархической структуры в приближении, когда параметр Ламэ $\mu=0$, как во включении, так и во вмещающей его среде. В этом случае динамиче-

ская задача сейсмологии может рассматриваться независимо для случая распространения продольной и поперечной волны. В данной работе будет рассмотрен первый случай для предложенной модели. Полученные результаты могут быть использованы для выбора критериев комплексирования сейсмических методов исследования сложно построенных сред.

1. Задача о дифракции звука в двумерной пористой влагонасыщенной неоднородности n -слойной среды

Эту задачу будем решать, используя подход, изложенный в работах [2–4]. Массовые силы Φ будем считать потенциальными и сосредоточенными в первом слое n -слойной среды. Плоскость XOY совпадает с верхней плоскостью 1-го слоя, $z=0$. Ось OZ направлена вертикально вниз. Образующие двумерной неоднородности в виде цилиндра произвольного сечения S_0 направлены вдоль оси OY . При $\mu=0$ в каждом из слоев S_i выполняется первое уравнение из системы уравнений для прямой динамической задачи сейсмологии [2], преобразованное к виду:

$$\begin{aligned} \Delta\varphi_i + k_{1i}^2\varphi_i &= -2\pi f_i(M); \\ \vec{u} = \text{grad}\varphi; f_i(M) &= \frac{\sigma_i}{2\pi\lambda_i} \Phi_i; \end{aligned} \quad (1)$$

где $i=1, \dots, n$, $\Phi_i=\Phi$ при $i=1$, при $i \neq 1$ $\Phi_i=0$. Волновое число в i -ом слое равно согласно [2–4]:

$$k_{1i}^2 = k_1^2 = \omega^2 \frac{\sigma_i}{\lambda_i}; \quad (2)$$

где ω — круговая частота, σ_i, λ_i — плотность и коэффициент Ламэ i -го слоя n -слойной среды. Пусть в J -м слое n -слойной упругой среды находится пористое влагонасыщенное включение (вода или нефть). Согласно [5], чтобы оставаться в равновесии жидкая фаза должна испытывать одинаковое во всех точках образуемого порами пространства гидростатическое давление p_2 . Это давление должно действовать и на вмещающую среду. Обусловленная им ее деформация должна сводиться к изменению объема фазы V_1 и пор V_2 в одном и том же отношении:

$$\frac{\Delta V_1}{V_1} = \frac{\Delta V_2}{V_2} = -\frac{1}{K_0} p_2, \quad (3)$$

K_0 — истинный модуль сжимаемости фазы. В том же отношении меняется и весь макроскопический объем среды. Для покоящейся в пористом включении жидкости внутри включения система (1), согласно [5] после несложных преобразований записывается в виде:

$$\begin{aligned} \Delta\varphi_i + k_{1i}^2\varphi_i &= -2\pi(f_i(M) + \psi); \\ \vec{u} = \text{grad}\varphi; f_i(M) &= \frac{\sigma_i}{2\pi\lambda_i} \Phi_i; \psi = (1 - \chi - \frac{K}{K_0})p_2; \end{aligned} \quad (4)$$

$$\psi = \begin{cases} \psi(M), M \in S_0 \\ 0, M \notin S_0 \end{cases} \quad (5)$$

$K = \lambda$ — модуль всестороннего сжатия при $\mu=0$, χ — пористость.

Введем обозначение:

$\tilde{k}(M)=k_{1i}$ — волновое число в слоистой среде S_i , $i=1, \dots, n$ и

$$K(M) = \begin{cases} k_{1ji} \text{ при } M \in S_0 \\ \tilde{k}(M) \text{ при } M \notin S_0 \end{cases} \quad k_{1ji}^2 = \omega^2 \frac{\sigma_{ji}}{\lambda_{ji}}; \quad (6)$$

Индекс ji обозначает свойства среды, внутри неоднородности S_0 . В общем случае в произвольном слое S_i или внутри неоднородности S_0 уравнение (4) с учетом (2), (5) и (6) будет иметь вид:

$$\Delta \varphi_i + k_{1i}^2 \varphi_i = -2\pi(f_i(M) + \psi); \quad (7)$$

Граничные условия в среде без разрывов заключающиеся в непрерывности вертикальной составляющей вектора смещения и компонентов тензора напряжений, согласно [2], на границах раздела L_i имеют вид:

$$\begin{aligned} \frac{\partial \varphi_i}{\partial z} - \frac{\partial \varphi_{i+1}}{\partial z} &= 0 \Big|_{z \in L_i}; \quad \left(\frac{\partial \varphi}{\partial n} \right)_{ji} = \left(\frac{\partial \varphi}{\partial n} \right)_{ja}; \\ \left[\sigma_i(\omega^2 \varphi_i + \Phi_i) \right] - \left[\sigma_{i+1}(\omega^2 \varphi_{i+1} + \Phi_{i+1}) \right] &= 0 \Big|_{z \in L_i}; \\ \left[\sigma(\omega^2 \varphi + \Phi + \alpha p_2) \right]_{ji} &= \left[\sigma(\omega^2 \varphi + \Phi) \right]_{ja}; \\ \left[\sigma(\omega^2 \varphi + \Phi) \right] \Big|_{z=0} &= 0; \end{aligned} \quad (8)$$

индекс ji — обозначает значения σ, φ, Φ на границе неоднородности с внутренней стороны, ja — с внешней стороны границы неоднородности, которая расположена в j -ом слое, L -граница раздела слоя, с индексом i — со стороны i -го слоя, с индексом $i+1$ — со стороны $(i+1)$ -го слоя, $\alpha = 1 - \chi - \frac{K}{K_0}$, согласно (4).

Условия затухания на бесконечности согласно [2] имеют вид:

$$r \operatorname{grad} \varphi_i = O(1), \quad r \left(\frac{\partial \varphi_i}{\partial r} - ik_1 \varphi_i \right) = o(1). \quad (9)$$

Пусть:

$$\tilde{\varphi}_i = \varphi_i - \varphi_i^0, \quad (10)$$

где $i=1, \dots, j, ji, \dots, n$, φ_i^0 - потенциал нормального сейсмического поля в слоистой среде в отсутствие неоднородности: $\varphi_{ji}^0 = \varphi_i^0$ и

$$\Delta \varphi_i^0 + k_{1i}^2 \varphi_i^0 = -2\pi f_i(M); \quad (11)$$

$$\frac{\partial \varphi_i^0}{\partial z} - \frac{\partial \varphi_{i+1}^0}{\partial z} = 0 \Big|_{z \in L_i}; \quad (12)$$

$$\left[\sigma_i (\omega^2 \varphi_i^0 + \Phi_i) \right] - \left[\sigma_{i+1} (\omega^2 \varphi_{i+1}^0 + \Phi_{i+1}) \right] = 0 \Big|_{z \in L_i}.$$

На контуре неоднородности φ^0 и $\frac{\partial \varphi^0}{\partial n}$ непрерывны. Алгоритм вычисления нормального поля при дифракции звука в слоистой среде для произвольного источника возбуждения изложен в работах [6–7].

$\tilde{\varphi}_i$ – потенциал аномального сейсмического поля, который, как легко показать с учетом (2),(4),(7) и (11) удовлетворяет уравнению:

$$\Delta \tilde{\varphi}_i + K^2(M) \tilde{\varphi}_i = -(K^2(M) - \tilde{k}^2(M))(\varphi_i^0 + \alpha p_2); \quad (13)$$

и граничным условиям:

$$\frac{\partial \tilde{\varphi}_i}{\partial z} - \frac{\partial \tilde{\varphi}_{i+1}}{\partial z} = 0 \Big|_{z \in L_i}; \left(\frac{\partial \tilde{\varphi}}{\partial n} \right)_{ji} = \left(\frac{\partial \tilde{\varphi}}{\partial n} \right)_{ja}; \quad (14)$$

$$\left[\sigma_i \omega^2 \tilde{\varphi}_i \right] - \left[\sigma_{i+1} \omega^2 \tilde{\varphi}_{i+1} \right] = 0 \Big|_{z \in L_i};$$

На контуре неоднородности:

$$[\sigma \tilde{\varphi}]_{ja} - [\sigma \tilde{\varphi}]_{ji} = (\sigma_{ja} - \sigma_{ji})(\varphi_j^0 + \alpha p_2) \quad (15)$$

Функция источника сейсмического поля $G_{Sp}(M, M^0)$ определяется как решение следующей краевой задачи [3–4]:

$$\Delta G_{Spi} + \tilde{k}^2 G_{Spi}(M, M^0) = -2\pi \delta(M - M^0); \quad (16)$$

и граничным условиям:

$$\frac{\partial G_{Sp,i}}{\partial z} - \frac{\partial G_{Sp,i+1}}{\partial z} = 0 \Big|_{z \in L_i}; \left[\sigma \omega^2 G_{Sp} \right] \Big|_{z=0} = 0; \quad (17)$$

$$\left[\sigma_i \omega^2 G_{Sp,i} \right] - \left[\sigma_{i+1} \omega^2 G_{Sp,i+1} \right] = 0 \Big|_{z \in L_i}.$$

На контуре неоднородности $G_{Sp}(M, M^0)$ и $\frac{\partial G_{Sp}(M, M^0)}{\partial n}$ непрерывны. Применим формулу Грина [2] для функций $\tilde{\varphi}_i$ и $G_{Sp}(M, M^0)$ для каждого слоя n -слойной среды при $i \neq j$:

$$\int_{L_i} (\tilde{\varphi}_i \frac{\partial G_{Sp,i}}{\partial n} - G_{Sp,i} \frac{\partial \tilde{\varphi}_i}{\partial n}) dl_i - \int_{L_{i+1}} (\tilde{\varphi}_{i+1} \frac{\partial G_{Sp,i+1}}{\partial n} - G_{Sp,i+1} \frac{\partial \tilde{\varphi}_{i+1}}{\partial n}) dl_{i+1} = \begin{cases} 2\pi \tilde{\varphi}_i(M^0) n p u M^0 \in S_i \\ 0 n p u M^0 \notin S_i \end{cases} \quad (18)$$

При $i=j$:

$$\int_{L_j} (\tilde{\varphi}_j \frac{\partial G_{Sp,j}}{\partial n} - G_{Sp,j} \frac{\partial \tilde{\varphi}_j}{\partial n}) dl_j - \int_{L_{j+1}} (\tilde{\varphi}_{j+1} \frac{\partial G_{Sp,j+1}}{\partial n} - G_{Sp,j+1} \frac{\partial \tilde{\varphi}_{j+1}}{\partial n}) dl_{j+1} + \oint_{c_{ja}} (\tilde{\varphi}_{ja} \frac{\partial G_{Sp,j}}{\partial n} - G_{Sp,j} \frac{\partial \tilde{\varphi}_{ja}}{\partial n}) dc_{ja} = \begin{cases} 2\pi \tilde{\varphi}_j(M^0) n p u M^0 \in S_j - S_0 \\ 0 n p u M^0 \notin S_j - S_0 \end{cases} \quad (19)$$

c_{ja} – контур неоднородности C (внешняя сторона).

Домножим выражения (18) и (19) на σ_i и сложим их с учетом граничных условий, в результате чего получим:

$$\frac{\sigma_j}{2\pi} \oint_c (\tilde{\varphi}_{ja} \frac{\partial G_{Sp,j}}{\partial n} - G_{Sp,j} \frac{\partial \tilde{\varphi}_{ja}}{\partial n}) dc = \begin{cases} \sigma(M^0) \tilde{\varphi}(M^0) n p u M^0 \in S_j - S_0 \\ 0 n p u M^0 \notin S_j - S_0 \end{cases} \quad (20)$$

Применим формулу Грина для функций $\tilde{\varphi}_{ji}$ и $G_{Sp,j}(M, M^0)$ для внутренности области S_0 , с учетом (4), (5), (11), (12), (13), (14), (15), (16) и (17) получим:

$$\frac{k_{1ji}^2 - k_{1j}^2}{2\pi} \iint_{S_0} \varphi(M) G_{Sp,j}(M, M^0) d\tau_M - \frac{1}{2\pi} \oint_c (\tilde{\varphi}_{ji} \frac{\partial G_{Sp,j}}{\partial n} - G_{Sp,j} \frac{\partial \tilde{\varphi}_{ji}}{\partial n}) dc = \begin{cases} (\tilde{\varphi}(M^0) + \alpha p_2) n p u M^0 \in S_0 \\ 0 n p u M^0 \notin S_0 \end{cases} \quad (21)$$

Домножим выражение (21) на σ_{ji} и сложим полученный результат с выражением (20). С учетом граничных условий на контуре неоднородности получим:

$$\frac{\sigma_{ji}(k_{1ji}^2 - k_{1j}^2)}{2\pi} \iint_{S_0} \varphi(M) G_{Sp,j}(M, M^0) d\tau_M + \frac{(\sigma_{ja} - \sigma_{ji})}{2\pi} \oint_c \varphi^0(M) \frac{\partial G_{Sp,j}}{\partial n} dc - \frac{(\sigma_{ja} - \sigma_{ji})}{2\pi} \oint_c G_{Sp,j} \frac{\partial \varphi}{\partial n} dc - \frac{(\sigma_{ja} - \sigma_{ji})}{2\pi} \oint_c G_{Sp,j} \frac{\partial \varphi^0}{\partial n} dc = \sigma(M^0) (\tilde{\varphi}(M^0) + \alpha p_2); \quad (22)$$

Воспользуемся равенством [2]:

$$\frac{(\sigma_{ja} - \sigma_{ji})}{2\pi} \oint_c (\varphi^0(M) \frac{\partial G_{Sp,j}}{\partial n} - G_{Sp,j} \frac{\partial \varphi^0(M)}{\partial n}) dc = \begin{cases} (\sigma_{ja} - \sigma_{ji}) \varphi^0(M^0) n p u M^0 \in S_0 \\ 0 n p u M^0 \notin S_0 \end{cases} \quad (23)$$

Тогда выражение (22) с учетом (23) можно переписать в виде:

$$\frac{(k_{1ji}^2 - k_{1j}^2)}{2\pi} \iint_{S_0} \varphi(M) G_{Sp,j}(M, M^0) d\tau_M + \frac{\sigma_{ja}}{\sigma_{ji}} \varphi^0(M^0) - \frac{(\sigma_{ja} - \sigma_{ji})}{\sigma_{ji} 2\pi} \oint_c G_{Sp,j} \frac{\partial \varphi}{\partial n} dc = (\varphi(M^0) + \alpha p_2) \text{ при } M^0 \in S_0 \quad (24)$$

$$\frac{\sigma_{ji}(k_{1ji}^2 - k_{1j}^2)}{\sigma(M^0) 2\pi} \iint_{S_C} \varphi(M) G_{Sp,j}(M, M^0) d\tau_M + \varphi^0(M^0) - \frac{(\sigma_{ja} - \sigma_{ji})}{\sigma(M^0) 2\pi} \oint_c G_{Sp,j} \frac{\partial \varphi}{\partial n} dc = \varphi(M^0) \text{ при } M^0 \notin S_0 \quad (24')$$

Таким образом, решив интегро-дифференциальное уравнение (24) и определив распределение потенциала вектора смещений внутри неоднородности, мы можем, используя второе интегро-дифференциальное представление (24'), определить потенциал вектора смещений в любом слое, а затем, используя соотношение (1), вычислить распределение вектора смещений в любом слое.

2. Сопоставление алгоритмов моделирования сейсмического поля для случаев упругого и пористого включения

Сравним полученные выражения с решением задачи дифракции сейсмического поля в рамках той же геометрической модели но с различными физическими свойствами включения. В работе [3] выписана система интегро-дифференциальных уравнений для случая упругого включения в n -слойной среде:

$$\frac{(k_{1ji}^2 - k_{1j}^2)}{2\pi} \iint_{S_0} \varphi(M) G_{Sp,j}(M, M^0) d\tau_M + \frac{\sigma_{ja}}{\sigma_{ji}} \varphi^0(M^0) - \frac{(\sigma_{ja} - \sigma_{ji})}{\sigma_{ji} 2\pi} \oint_c G_{Sp,j} \frac{\partial \varphi}{\partial n} dc = \varphi(M^0) \text{ при } M^0 \in S_0$$

$$\frac{\sigma_{ji}(k_{1ji}^2 - k_{1j}^2)}{\sigma(M^0) 2\pi} \iint_{S_C} \varphi(M) G_{Sp,j}(M, M^0) d\tau_M + \varphi^0(M^0) - \frac{(\sigma_{ja} - \sigma_{ji})}{\sigma(M^0) 2\pi} \oint_c G_{Sp,j} \frac{\partial \varphi}{\partial n} dc = \varphi(M^0) \text{ при } M^0 \notin S_0 \quad (25)$$

Обозначения в (25) те же, что и в (24) и (24'). Сравнивая (25) и (24), легко заметить, что имеет место различие в структуре свободного члена в интегро-дифференциальном уравнении для внутренней задачи. Это, безусловно, скажется на различии в решении внешней задачи для этих двух моделей. Однако наличие пористого включения не приводит к изменению волнового числа для рассматриваемой задачи распространения продольной волны в рамках двух разных моделей среды, что может свидетельствовать о

неинформативности кинематических характеристик продольных волн для идентификации пористого влагонасыщенного включения.

3. Моделирование дифракции звука на упругой неоднородности иерархического типа с пористым включением

Рассмотрим задачу дифракции звука на двумерной упругой неоднородности с иерархической структурой, расположенной в j -ом слое n -слойной среды [4]. Если при переходе на следующий иерархический уровень ось двухмерности не меняется, а меняются только геометрии сечений вложенных структур, то аналогично (1) можно выписать итерационный процесс моделирования сейсмического поля (случай формирования только продольной волны).

Идею, изложенную в работе [4] для решения прямой задачи для двумерного случая распространения продольной волны через локальную упругую неоднородность с иерархической структурой, расположенной в J -ом слое n -слойной среды, распространим на случай, когда на L -ом иерархическом уровне окажется пористое влагонасыщенное включение.

$$\begin{aligned} & \frac{(k_{1jil}^2 - k_{1j}^2)}{2\pi} \iint_{S_{Cl}} \varphi_l(M) G_{Sp,j}(M, M^0) d\tau_M + \frac{\sigma_{ja}}{\sigma_{jil}} \varphi_{l-1}^0(M^0) - \\ & - \frac{(\sigma_{ja} - \sigma_{jil})}{\sigma_{jil} 2\pi} \oint_c G_{Sp,j} \frac{\partial \varphi_l}{\partial n} dc = \varphi_l(M^0), M^0 \in S_{Cl} \\ & \frac{\sigma_{jil}(k_{1jil}^2 - k_{1j}^2)}{\sigma(M^0) 2\pi} \iint_{S_{Cl}} \varphi_l(M) G_{Sp,j}(M, M^0) d\tau_M + \varphi_{l-1}^0(M^0) - \\ & - \frac{(\sigma_{ja} - \sigma_{jil})}{\sigma(M^0) 2\pi} \oint_c G_{Sp,j} \frac{\partial \varphi_l}{\partial n} dc = \varphi_l(M^0), M^0 \notin S_{Cl} \end{aligned} \quad (26)$$

где $G_{Sp,i}(M, M^0)$ — функция источника сейсмического поля, она совпадает с функцией выражения (16), (17), $k_{1jil}^2 = \omega^2(\sigma_{jil} / \lambda_{jil})$ — волновое число для продольной волны. В приведенном выражении индекс ji обозначает принадлежность свойств внутри неоднородности, ja — вне неоднородности, $l=1...L-1$ — номер иерархического уровня, $\vec{u}_l = grad \varphi_l$, φ_l^0 — потенциал нормального сейсмического поля в слоистой среде в отсутствие неоднородности предыдущего ранга, если $l=2...L$, $\varphi_l^0 = \varphi_{l-1}$, если $l=1$, $\varphi_l^0 = \varphi^0$, что совпадает с соответствующим выражением из [4]. Если при переходе на следующий иерархический уровень ось двухмерности не меняется, а меняются только геометрии сечений вложенных структур, то аналогично [4] можно выписать итерационный процесс моделирования сейсмического поля (случай формирования только продольной волны). Итерационный процесс относится к моделированию вектора смещений при переходе с предыдущего иерархического уровня на последующий уровень. Внутри каждого иерархического уровня интегро-дифференциальное уравнение и интегро-дифференциальное

представление выписаны для потенциала, через которого выражается вектор смещений (26). Если на некотором иерархическом уровне структура локальной неоднородности распадается на несколько неоднородностей, то двойной и контурные интегралы в выражениях (26) берутся по всем неоднородностям. Если $l=L$, то внутри неоднородностей предыдущего иерархического уровня оказывается пористая влагонасыщенная неоднородность. В этом случае система (26) с учетом (24) и (24') переписывается в виде:

$$\begin{aligned} & \frac{(k_{1jil}^2 - k_{1j}^2)}{2\pi} \iint_{S_{0l}} \varphi_l(M) G_{Sp,j}(M, M^0) d\tau_M + \frac{\sigma_{ja}}{\sigma_{jil}} \varphi_{l-1}^0(M^0) - \\ & - \frac{(\sigma_{ja} - \sigma_{jil})}{\sigma_{jil} 2\pi} \oint_c G_{Sp,j} \frac{\partial \varphi_l}{\partial n} dc = (\varphi_l(M^0) + \alpha p_2), M^0 \in S_{0l} \end{aligned} \quad (27)$$

$$\begin{aligned} & \frac{\sigma_{jil}(k_{1jil}^2 - k_{1j}^2)}{\sigma(M^0) 2\pi} \iint_{S_{0l}} \varphi_l(M) G_{Sp,j}(M, M^0) d\tau_M + \varphi_{l-1}^0(M^0) - \\ & - \frac{(\sigma_{ja} - \sigma_{jil})}{\sigma(M^0) 2\pi} \oint_c G_{Sp,j} \frac{\partial \varphi_l}{\partial n} dc = \varphi_l(M^0), M^0 \notin S_{0l}, l = L \end{aligned} \quad (27')$$

Если $l=L+1$ и на следующем уровне неоднородность снова упругая, то для дальнейшего продолжения итерационного процесса мы снова можем использовать выражения (26).

Заключение

В работе рассмотрена задача моделирования сейсмического поля (случай распространения продольной волны с учетом только сжатия и растяжения среды) в слоистой среде с включениями иерархической структуры. Построены алгоритмы моделирования в сейсмическом случае для 2-D упругой и пористой неоднородности. Исследован вопрос отражения физических свойств однофазности и двухфазности в системе уравнений решения прямой динамической сейсмологии в частотном варианте. Представляет интерес с использованием полученных алгоритмов исследовать вопрос об изучении связи между тензорами напряжения и деформации на каждом иерархическом уровне и о возможном отклонении ее от обобщенного закона Гука. С другой стороны с увеличением степени иерархичности среды увеличивается степень пространственной нелинейности распределения составляющих вектора смещений, что предполагает исключение методов линеаризации задачи при создании методов интерпретации данных динамической сейсмологии. Кроме того, усложняется процесс комплексирования методов, использующих волновые поля для изучения отклика среды с иерархической многофазной структурой. Эта проблема неразрывно связана с формулировкой и решением обратной задачи для распространения электромагнитного и сейсмического полей в таких сложных средах, что является нашей ближайшей задачей.

Литература

1. Хасанов, М.М. Нелинейные и неравновесные эффекты в реологически сложных средах / М.М. Хасанов, Г.Т. Булгакова. — Москва — Ижевск: Институт компьютерных исследований, 2003. — 288 с.
2. Купрадзе, В.Д. Граничные задачи теории колебаний и интегральные уравнения / В.Д. Купрадзе — М. — Ленинград: Гос. Изд-во технико-теоретической литературы, 1950. — 280 с.
3. Хачай, О.А. О комплексировании сейсмических и электромагнитных активных методов для картирования и мониторинга состояния двумерных неоднородностей в N-слоистой среде / О.А. Хачай, А.Ю. Хачай // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». — 2011. — № 2(219), Вып. 13. — С. 49–56.
4. Хачай, О.А. Моделирование электромагнитного и сейсмического поля в иерархически неоднородных средах / О.А. Хачай, А.Ю. Хачай // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». — 2013. — Т. 2, № 2, — С. 48–55.
5. Френкель, Я.И. К теории сейсмических и сейсмоэлектрических явлений во влажной почве / Я.И. Френкель // Известия АН СССР, серия географическая и геофизическая. — 1944. — Т. 8., № 4, — С. 133–150.
6. Хачай, А.Ю. Алгоритм решения прямой динамической задачи сейсмологии при возбуждении горизонтальной точечной силой, расположенной в произвольном слое n-слоистой упругой изотропной среды / А.Ю. Хачай // Информатика и математическое моделирование, УрГУ, Екатеринбург. — 2006. — С. 170–278.
7. Хачай А.Ю. Алгоритм решения прямой динамической задачи сейсмологии при возбуждении точечным источником вертикальной силы, расположенной в произвольном слое n-слоистой упругой изотропной среды / А.Ю. Хачай // Информатика и математическое моделирование. УрГУ, Екатеринбург. — 2006. — С. 279–310.

Хачай Ольга Александровна, д.ф.-м.н., ведущий научный сотрудник, Институт геофизики УрО РАН, Екатеринбург olgakhachay@yandex.ru

Хачай Андрей Юрьевич, к.ф.-м.н., доцент, Уральский Федеральный Университет, Екатеринбург, andrey.khachay@usu.ru

Поступила в редакцию 20 декабря 2013 г.

MODELING OF SEISMIC FIELDS IN HIERARCHIC HETEROGENEOUS TWO PHASE MEDIA

O.A. Hachay, Institute of Geophysics UB RAS (Yekaterinburg, Russian Federation)

A.Y. Khachay, Ural Federal University (Yekaterinburg, Russian Federation)

In the paper is considered a problem for modeling seismic fields (case of longitudinal wave distribution with account of medium of compression and tension) in a layered medium with inclusions of hierarchic structure. It had been constructed an algorithms for 2D modeling of sound diffraction on porous fluid saturated intrusion of hierarchic elastic structure, located in the layer number J of N -layered elastic medium. It is analyzed the problem of reflection of physical features of one phase and two phase inclusions in the solution equations of the straight dynamical problem in a frequency variant.

Keywords: hierarchic medium, elastic, porous, two phase, seismic field, algorithms of modeling.

References

1. Hasanov M.M., Bulgakova G.G. Nelineinie I neravnovesnie effecti v reologicheski sloznych sredach [Nonlinear and non-equilibrium effects in rheological complicated Media]. Moscow, Izevsk, Publishing of the Institute of computing research, 2003. 288 p.
2. Kupradze V.D. Granichnie zadachi teorii kolebanij I integralnie uravnenija [Boundary problems of the theory of oscillations and integral equations]. Moscow, Leningrad, Publishing in "Izdatelstvo tehniko-teoreticheskoy literaturi", 1950. 280 p.
3. Hachay O.A., Khachay A.Y. O kompleksirovani seismicheskikh I elektromagnitnich aktivnich metodov dlja kartirovaniya I monitoringa sostojaniya dvumernich neodnorodnostey v N -sloinoy srede [About Integrating Seismic and Electromagnetic Active Methods for Mapping and Monitoring of the State of 2-D Heterogeneous Objects in N -layered Medium]. Vestnik Yuzhno-Uralskogo gosudarstvennogo universiteta. Seriya "Kompyuternije tehnologii, upravlenije, radioelektronika" [Bulletin of South Ural State University. Series "Computer technologies, control, radioelectronics"], 2011. No 2 (219). P. 49–56.
4. Hachay O.A., Khachay A.Y. Modelirovaniye electromagnitnogo I seismicheskogo polja v ierarchicheski neodnorodnich sredach. [Modelling of electromagnetic and seismic fields in hierarchic heterogeneous media]. Vestnik Yuzhno-Uralskogo gosudarstvennogo universiteta. Seriya "Vichislitel'naja matematika I informatika" [Bulletin of South Ural State University. Series: "Computational mathematics and Software Engineering"], 2013. V. 2, No 2, P. 48–55.

5. Frenkel Ya.I. K teorii seismicheskikh i seismoelektricheskikh yavlenij vo vlaznoi pochve [To the theory of seismic and seismoelectric effects in a humide soil], *Izvestija AN USSR*, 1944. V. 8, No 4, P. 133–150.
6. Khachay A.Y. Algorithm reshenija prjamoj dinamicheskoy zadachi seismiki pri vzbuzdenii gorizontальной tochečnoj siloj, raspolozhennoj v proizvolnom sloje n-slojnoj uprugoj izotropnoj sredi [Algorithm of solution of straight dynamical seismic problem by excitation of horizontal local force, located in an arbitrary layer of an n-layered elastic isotropic media]. *Informatika I matematicheskije modelirovanije* [Informatics and mathematical modeling]. Ekaterinburg, Publishing in the Ural State University, 2006. P. 170–278.
7. Khachay A.Y. Algorithm reshenija prjamoj dinamicheskoy zadachi seismiki pri vzbuzdenii tochečnim istočnikom vertikalnoj sili, raspolozhennoj v proizvolnom sloje n-slojnoj uprugoj izotropnoj sredi [Algorithm of solution of straight dynamical seismic problem by excitation of a local source of verical force, located in an arbitrary layer of an n-layered elastic isotropic media]. *Informatika I matematicheskije modelirovanije* [Informatics and mathematical modeling]. Ekaterinburg, Publishing in the Ural State University, 2006. P. 279–310.

Received 20 December 2013

РАЗРАБОТКА СИСТЕМЫ ДИНАМИЧЕСКОГО ИЗМЕНЕНИЯ КОНТЕНТА ВИДЕОИГР НА ОСНОВЕ ЭВОЛЮЦИОННОГО МОДЕЛИРОВАНИЯ

М.Г. Меженин

Темой данного исследования является применение методов эволюционного моделирования (ЭМ) для автоматического создания и динамического изменения мультимедийных данных в рамках процедурной генерации контента (ПГК) видеоигр. Использование таких методов ЭМ, как генетические алгоритмы, позволяет решить проблемы персонализации и адаптируемости контента, имеющую большую значимость в индустрии видеоигр. В работе приведен обзор существующих систем ПГК на основе ЭМ и предложен новый универсальный метод представления персонажей в подобных системах. В рамках исследования реализована система эволюционного моделирования, основанная на предложенном методе; в отличие от существующих решений, разработанная система является универсальной и может использоваться в различных коммерческих и научных проектах.

Ключевые слова: процедурная генерация контента, эволюционное моделирование, персонализация контента.

Введение

В современном мире индустрия электронных развлечений является одной из наиболее быстроразвивающихся в глобальной экономике; по различным оценкам, объем рынка видеоигр составляет от 50 до 70 миллиардов долларов. Все большой интерес к этой области появляется и у научного сообщества: создаются учебные направления, научные ассоциации, конференции и журналы. Одним из наиболее перспективных направлений исследований в этой сфере является процедурная генерация контента (ПГК), под которой понимают автоматическое алгоритмическое создание и динамическое изменение различных составляющих частей игр [11], а также правил и целей самого игрового процесса [3].

Использование ПГК позволяет не только значительно понизить стоимость создания контента, но и решает проблему персонализации, приобретающую большую значимость в связи с увеличением количества потенциальных игроков. Однако, в большинстве систем ПГК используются специализированные алгоритмы, применение которых ограничено параметрами одной конкретной игры [9]. Эту проблему можно эффективно решить с помощью методов эволюционного моделирования (ЭМ) [11, 12]. Таким образом, актуальной является задача разработки эволюционных моделей игрового контента различных видов и реализации на их основе универсальных систем ПГК.

В научной среде интерес к проблеме процедурной генерации контента в видеоиграх появился достаточно недавно. Теоретической базой данного исследования являются работы зарубежных ученых Д. Тогелиуса [11], Г. Яннакакиса [12] и М. Хендрикса [4] по классификации подходов к ПГК в видеоиграх.

В данной статье описывается разработка универсальной системы динамического изменения виртуальных персонажей на основе ЭМ для использования в различных научно-исследовательских и коммерческих игровых проектах и предложен новый метод представления персонажей игр в подобных системах.

Статья организована следующим образом. В первом разделе представлены историческая справка и обзор существующих подходов к ПГК в различных коммерческих и научно-исследовательских игровых проектах. Во втором разделе описан разработанный метод представления игровых персонажей в системах ПГК. В третьем разделе приведены детали проектирования и реализации системы динамического изменения контента на основе данного метода. В четвертом разделе описана апробация полученных результатов на основе разработанного прототипа видеоигры. В заключении суммируются основные результаты работы и рассматриваются направления дальнейших исследований.

1. Процедурная генерация контента на основе эволюционного моделирования

Под процедурной генерацией контента (ПГК) в видеоиграх понимается автоматическое создание и динамическое изменение различного игрового контента с помощью некоторого алгоритма [11]. Среди преимуществ ПГК отмечаются экономия средств, затраченных на производство, качественное повышение разнообразия и продолжительности игр, уменьшение требований к памяти устройств, а также возможность создания нового и необычного контента и даже совершенно новых игровых жанров, основанных на ПГК.

В той или иной мере, ПГК используется в видеоиграх с 1980-х годов; среди первых игр, важную роль в которых играл именно динамически генерируемый контент, можно отметить проекты *Rogue* (Toy and Wichma, 1980) и *Elite* (Braben and Bell, 1984). В настоящее время ПГК повсеместно используется в крупных коммерческих игровых проектах различных жанров, среди которых можно отметить стратегическую серию *Civilization* (Firaxis Games, 1991-2012), серию шутеров *Borderlands* (Gearbox Software, 2010-2012) и серию ролевых игр *Diablo* (Blizzard Entertainment, 1996-2013). Кроме того, ПГК используется в многочисленных играх от независимых разработчиков, например, в крайне популярной игре с открытым миром *Minecraft* (Mojang, 2009).

В современных коммерческих видеоиграх процедурная генерация используется для динамического создания различных видов контента: уровней, карт и ландшафтов, противников, снаряжения, а также игровых заданий. Несмотря на это, подход к генерации контента практически не изменялся; большинство реализаций основано на генераторе случайных чисел и не учитывает предпочтения пользователей или какие-либо другие факторы [5].

В качестве альтернатив классическому подходу исследователями предлагаются различные методы [4], однако только эволюционное моделирование (ЭМ) является достаточно гибким для представления различных видов игрового контента. Кроме того, ЭМ обеспечивает не только генерацию, но и адаптируемость контента к различным игрокам за счет эволюционного подхода. Общая схема работы алгоритмов ЭМ изображена на рис. 1.



Рис. 1. Схема работы алгоритмов эволюционного моделирования

В рамках ЭМ, эволюция может быть описана как многоступенчатый итерационный процесс, состоящий из случайных изменений в наборе изначально случайных решений («особей»), комбинирования («скрещивания») полученных результатов и последующего отбора лучших решений в новый набор («популяцию») на основе значений некоторой заданной функции оценки.

Использование методов ЭМ в системах ПГК отличается от классических алгоритмов следующими двумя ключевыми особенностями:

1) в системах ПГК используется интерактивная функция оценки, которая напрямую или опосредованно оценивает некоторый процесс взаимодействия с пользователями, а не сами особи;

2) целью ПГК в играх является не поиск некоторого единственного «оптимального» решения, а создание нового, разнообразного контента, способного увлекать игроков, в виду чего процесс генерации особей не ограничен по времени или итерациям.

Одним из главных факторов успешности эволюционного моделирования является выбор адекватного способа представления моделируемого контента, поскольку именно от него во многом зависит реализация алгоритма процедурной генерации, а также разнообразие и характеристики генерируемого контента [12]. Существуют различные способы представления контента в эволюционных алгоритмах, для каждого из которых характерны определенные типы генерируемого контента.

Наиболее распространенными и хорошо изученными являются различные способы представления контента на основе *векторов значений*, которые чаще всего используются для генерации игровых уровней [2]. В данном подходе каждое значение (представленное двоичным, целым или числом с плавающей запятой) соответствует некоторому параметру процедурного генератора.

Для представления контента в эволюционной модели могут использоваться различные виды *формальных грамматик*, которые задают алгоритм генерации контента и изменяются в процессе эволюции. Данный метод также используется для генерации игровых уровней [8].

Эволюционное моделирование на основе *искусственных нейронных сетей* является достаточно новым подходом и чаще всего используется для реализации адаптирующегося ИИ [10], однако существуют экспериментальные реализации данного метода для генерации графики и эффектов [5, 7]. Каждая особь в этом подходе представлена искусственной нейронной сетью, характеристики и структура которой изменяется в процессе эволюции.

Представление контента в виде *дерева* позволяет эволюционным моделям работать со сложными иерархическими видами контента, но вместе с тем значительно усложняет реализацию скрещивания из-за необходимости учета совместимости вершин в деревьях различной структуры. На основе данного метода была реализована система генерации трехмерных зданий [6].

Отличительным для игровой области является представление контента с помощью *структурных элементов*. Под структурными элементами понимаются небольшие логические компоненты, из которых состоит экземпляр контента. Каждый структурный элемент представляет собой кортеж значений разных типов, которые характеризуют данный компонент. Данный метод был предложен в рамках разработки универсального

подхода к процедурной генерации игровых уровней [9] и не использовался для представления других видов контента.

Таким образом, существующие решения в области процедурной генерации игрового контента на основе эволюционного моделирования можно разделить на пять основных видов в соответствии с используемым в них типом представления контента. Наиболее распространенным из них является класс представлений с помощью вектора значений, в то время как одним из самых перспективных является класс объектно-ориентированных представлений на основе структурных элементов.

Отметим, что в существующих научных работах не предпринималось попыток использовать системы ЭМ для моделирования характеристик игровых персонажей. По этой причине для данного вида контента необходима разработка соответствующего представления, учитывающего все параметры моделируемого контента. Проектирование такого представления описано в следующем разделе.

2. Разработка метода представления персонажей игр в системах ЭМ

Поиск наиболее подходящих способов представления различных видов контента является одной из главных проблем в области процедурной генерации [12]. Для определения требований к способу представления виртуальных персонажей в системе ЭМ их типичные характеристики были выявлены на основе анализа современных коммерческих видеоигр с динамической генерацией противников. В рассмотренных проектах персонажи чаще всего определяются тремя основными составляющими: числовыми характеристиками, способностями и категориями. Типичная структура, описывающая персонажей видеоигр, представлена на рис. 2.



Рис. 2. Структура виртуальных персонажей

На основе выявленных выше общих для виртуальных персонажей свойств можно сделать вывод, что данный вид игрового контента является комплексным и состоит из разнородных элементов. Ввиду этого факта целесообразным является разработка специализированного способа представления, который бы учитывал особенности структуры данного контента:

1) основные характеристики персонажей следует кодировать в виде вектора, для каждой из координат которого необходимо хранить название характеристики, используемый тип данных и область допустимых значений;

2) категорию персонажа нужно представлять в виде задаваемого пользователем перечислимого типа, значения которого интерпретируются в системе как целые числа;

3) для репрезентации способностей рационально использовать представление на основе структурных элементов, в которых тип взаимодействия, его объект и характеристика кодируются с помощью перечислимых типов, а значение и продолжительность изменения — с помощью числовых значений.

На рис. 3 приведен пример описания противника «Радскорпион» из игры Fallout 3 с помощью данного представления.

<p>Характеристики (здоровье: 150, защита: 0.0, восприятие: 4, агрессивность: 2, уверенность: 4, взаимопомощь: 2, отравление: 0, ...)</p> <p>Категория (тип: зверь)</p> <p>Способность (тип: атака, объект: противники, характеристика: отравление, значение: 3, длительность: 5)</p>

Рис. 3. Пример представления противника из игры Fallout 3

Как видно из приведенного примера, разработанное представление может быть успешно использовано для репрезентации виртуальных персонажей различных видов в игровых системах ЭМ. Подобное представление позволяет разделять различные параметры персонажей и определять для каждого из них различные операторы скрещивания и мутации. Одновременно с этим, между ними сохраняется связь, которая обеспечивает простоту работы со всеми свойствами объекта целиком.

3. Разработка системы эволюционного моделирования

На основе описанного способа представления в рамках данной работы была разработана система эволюционного моделирования и динамического изменения виртуальных персонажей видеоигр. Общая схема работы разработанной системы изображена на рис. 4.

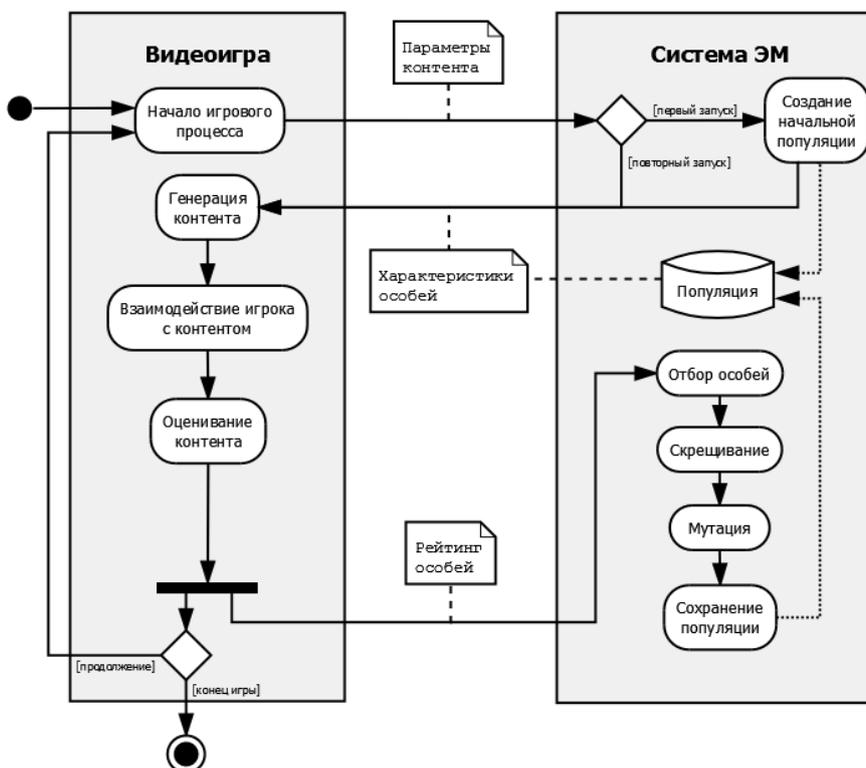


Рис. 4. Схема взаимодействия системы изменения контента с видеоигрой

Данная система изменения контента не зависит от реализации конкретной игры. Кроме того, она обеспечивает возможность гибкого задания параметров генерируемого контента во внешнем хранилище данных и может использоваться в качестве компонента различных игровых проектов.

По необходимости видеоигра запрашивает у системы ЭМ параметры текущей популяции, и на их основе генерирует виртуальных персонажей. Сгенерированный контент предоставляется игроку и, по результатам игровой сессии, каким-либо способом оценивается. Функция оценки на данном этапе может быть произвольной — например, пропорциональной нанесенному противниками урона игроку, некоторым образом интерпретирующей поведение игрока, и так далее.

Сведения об оценке каждого экземпляра популяции передаются обратно в систему ЭМ, в которой на основе полученных данных реализуются операции отбора, скрещивания и мутации. Результатом этого процесса становится новая популяция особей, которая может быть вновь передана видеоигре, либо сохранена во внешнее хранилище для последующей работы.

Ввиду использования в эволюционной модели нового способа представления контента, особый интерес представляет проектирование эволюционного алгоритма, а именно способ реализации в нем операций отбора, скрещивания и мутации особей.

Операция отбора реализована на основе метода турнирной селекции с копированием лучших особей в новую популяцию. В отличие от других методов, турнирная селекция не только гарантирует неубывание суммарной пригодности популяции, но и предотвращает преждевременное удаление потомков, способствуя разнообразию популяции [1]. Кроме того, поскольку на обучение игроков новым противникам тратится некоторое время, изменение всех особей популяции на каждом шаге нежелательно и целесообразным является использование частичной замены популяции.

Скрещивание особей происходит на основе попарного однородного кроссовера каждого из параметров родителей. Данный метод заключается в том, что значение каждой характеристики потомка с равной вероятностью копируется из характеристики одного из родителей. Способности потомка также выбираются случайным образом из способностей предков, общим числом не превышая максимального количества способностей предков.

Операция мутации для числовых характеристик основана на распределении Гаусса и заключается в изменении значения характеристики в соответствии со следующей формулой:

$$\begin{aligned} A' &= A + \Delta \times step, \quad \Delta \in \{-3; -2; -1; 1; 2; 3\}, \\ P(3) &= P(-3) = 0,021, \\ P(2) &= P(-2) = 0,136, \\ P(1) &= P(-1) = 0,341, \end{aligned} \quad (1)$$

где A — значение характеристики до мутации; A' — новое значение характеристики; $step$ — минимальное изменение характеристики; Δ — множитель, значения которого определяются в соответствии с плотностями вероятности нормального распределения.

4. Интеграция и проверка адекватности системы

Для проверки адекватности разработанной системы и реализованных в ней алгоритмов, на ее основе был создан игровой прототип. Как уже было отмечено, разрабо-

танная система является универсальной и может быть интегрирована в существующие проекты.

Разработанный прототип представляет собой видеоигру, целью которой является набор очков, которые даются за убийство противников и снимаются при получении урона. Так как игровой процесс акцентирован именно на противниках, он хорошо подходит для тестирования системы.

Персонажи видеоигры характеризуются семью характеристиками. Один игровой уровень состоит из нескольких последовательных «волн» противников. Каждая волна представляет собой одну популяцию в эволюционной модели, и генерируется после уничтожения предыдущей. Размер популяции эволюционной модели зависит от конкретного уровня, количество особей в селекционном турнире равно 4, вероятность мутации — 30 %.

Для ранжирования сложности и интересности противников была разработана функция оценки, которую целесообразно использовать в играх рассматриваемого жанра. Данная функция оценивает противника тем выше, чем большую опасность он представлял (т.е. чем больше урона он нанес, а также насколько он успел приблизиться), а также негативно оценивает гибель игрока. Описанная функция имеет следующий вид:

$$f = \begin{cases} -r + k \times d, & \text{игрок выжил} \\ -r - k \times d, & \text{игрок погиб} \end{cases} \quad (2)$$

где r — расстояние от противника до игрока в момент гибели противника; d — нанесенный противником урон; k — коэффициент, задающий отношение между единицами измерения расстояния и урона.

Разработанная игра была предложена пятнадцати игрокам с различным уровнем умений. Для проверки возможности адаптации системы из различных начальных условий, каждому человеку было предложено пройти пять уровней различной сложности.

Нами было определено два основных критерия, по которым можно судить об адекватности разработанной системы. Продемонстрируем их выполнение на примерах.

По графику на рис. 5 видно, что на первом, легком уровне, данный игрок стабильно набирает очки, и сложность противников увеличивается. В то же время на четвертом уровне игрок теряет очки, и система снижает сложность противников. Таким образом, существует прямая зависимость между успешностью игры пользователей и сложностью генерируемых противников.

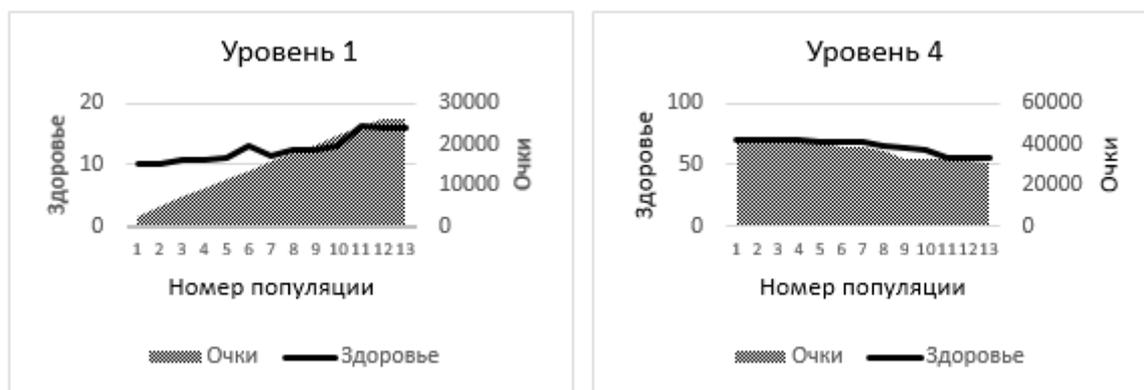


Рис. 5. Динамика изменения среднего значения характеристики «здоровье»

По графикам на рис. 6 видно, что в начале рассматриваемого уровня сложность обеспечивается высоким значением здоровья; в середине — низким здоровьем вместе с растущей силой; в конце — снова высоким здоровьем и снижающейся силой. Таким образом, мы показали наличие неоднородности и разнообразия изменений особей в различных популяциях ввиду наличия нескольких характеристик, влияющих на итоговую сложность персонажа.

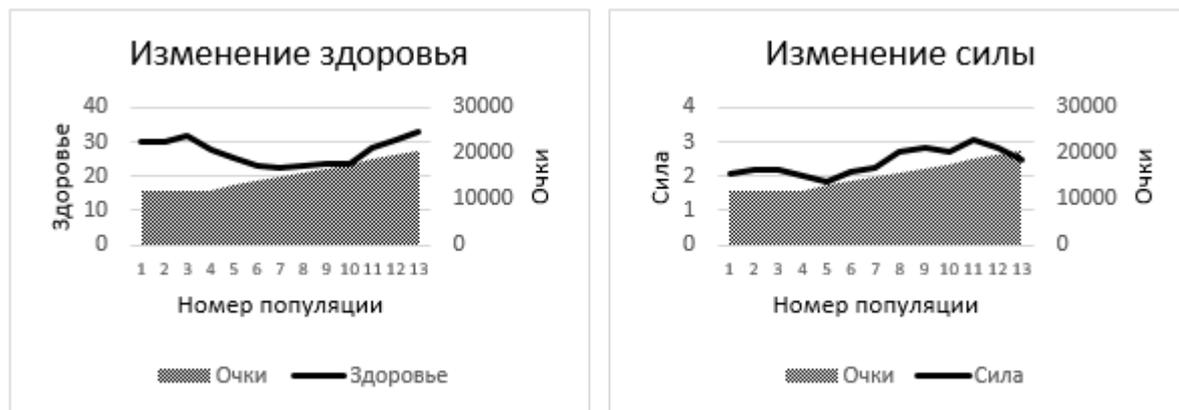


Рис. 6. Динамика изменения среднего значения характеристик «здоровье» и «сила»

На основе проведенного анализа результатов проверки адекватности можно утверждать, что разработанная система адекватно и корректно реализует требуемую функциональность. Характеристики персонажей успешно адаптируются к поведению пользователей, что подтверждается как статистическими данными, так и отзывами игроков. Использование эволюционного моделирования обеспечивает плавность изменения контента, одновременно с этим способствуя его разнообразию.

Заключение

Данная статья посвящена исследованию возможности использования эволюционного моделирования для динамического адаптивного изменения виртуальных персонажей в видеоиграх на основе предпочтений и навыков пользователей.

В статье предложен способ представления игровых персонажей в системах эволюционного моделирования и описана разработанная на его основе универсальная система процедурной генерации контента. Для проверки корректности реализованной системы была проведена проверка ее адекватности, результаты которой продемонстрировали корректность решения рассматриваемой проблемы.

В дальнейших исследованиях планируется в следующих основных направлениях:

- 1) использование разработанного способа представления для кодирования в эволюционных моделях информации о других игровых объектах, в частности — предметах снаряжения, разделяющих с игровыми персонажами большинство свойств;
- 2) изучение возможности добавления в разработанную систему дополнительных средств задания ограничений на генерируемый контент;
- 3) реализация на основе данной работы многопользовательской игровой системы, агрегирующей результаты взаимодействия всех пользователей для генерации.

Литература

1. Ashlock, D. Evolutionary Computation for Modeling and Optimization / D. Ashlock — New York: Springer, 2006. — 571 p.
2. Ashlock, D. Search-Based Procedural Generation of Maze-Like Levels / D. Ashlock // IEEE Transactions on Computational Intelligence and AI in Games. — Sept. 2011. — Vol. 3, No 3. — P. 260–273.
3. Cook, M. Multi-Faceted Evolution of Simple Arcade Games / M. Cook, S. Coulton // IEEE Conference on Computational Intelligence and Games. — Aug. 2011. — P. 289–296.
4. Hendrikx, M. Procedural Content Generation for Games: A Survey / M. Hendrikx, S. Meijer, J. Van Der Velden, A. Iosup // ACM Transactions on Multimedia Computing, Communications, and Applications. — Feb. 2013. — Vol. 9, No 1. — P. 1–22.
5. Hastings, E.J. Automatic Content Generation in the Galactic Arms Race Video Game / E.J. Hastings, R.K. Guha, K.O. Stanley // IEEE Transactions on Computational Intelligence and AI in Games. — Dec. 2009. — Vol. 1, No 4. — P. 245–263.
6. Martin, A. Evolving 3D Buildings for the Prototype Video Game Subversion / A. Martin, A. Lim, S. Colton, C. Browne // Proceedings of the 2010 International Conference on Applications of Evolutionary Computation. — Nov. 2010. — Vol. 6024. — P. 111–120.
7. Risi, S. Combining Search-based Procedural Content Generation and Social Gaming in the Petalz Video Game / S. Risi, J. Lehman, D.B. D'Ambrosio, R. Hall, K.O. Stanley // Proceedings of the 8th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. — Oct. 2012. — P. 63–68.
8. Shaker, N. Evolving Levels for Super Mario Bros Using Grammatical Evolution / N. Shaker, G.N. Yannakakis, J. Togelius, M. Nicolau, M. O'Neill // IEEE Conference on Computational Intelligence and Games. — Sept. 2012. — P. 304–311.
9. Sorenson, N. A Generic Approach to Challenge Modeling for the Procedural Creation of Video Game Levels / N. Sorenson, P. Pasquier, S. DiPaola // IEEE Transactions on Computational Intelligence and AI in Games. — Sept. 2011. — Vol. 3, No 3. — P. 229–244.
10. Stanley, K.O. Real-Time Neuroevolution in the NERO Video Game / K.O. Stanley, B.D. Bryant, R. Miikkulainen // IEEE Transactions on Evolutionary Computation. — Dec. 2005. — Vol. 9, No 6. — P. 653–668.
11. Togelius, J. Search-Based Procedural Content Generation: A Taxonomy and Survey / J. Togelius, G.N. Yannakakis, K.O. Stanley, C. Browne // IEEE Transactions on Computational Intelligence and AI in Games. — Sept. 2011. — Vol. 3, No 3. — P. 172–186.
12. Yannakakis, G.N. Experience-Driven Procedural Content Generation / G.N. Yannakakis, J. Togelius // IEEE Transactions on Affective Computing. — July 2011. — Vol. 2, No 3. — P. 147–161.

Меженин Михаил Григорьевич, аспирант, лаборант кафедры системного программирования, Южно-Уральский государственный университет (Челябинск, Российская Федерация), m.mezhenin@gmail.com.

Поступила в редакцию 25 декабря 2013 г.

EVOLUTIONARY-BASED PROCEDURAL CONTENT GENERATION FOR VIDEOGAMES

M.G. Mezhenin, South Ural State University (Chelyabinsk, Russian Federation)

The focus of this research is applying evolutionary algorithms to automatically generating and dynamically changing multimedia data within the context of procedural content generation (PCG) for videogames. Evolutionary computing allows personalizing and adapting the generated content, which is an increasingly important issue in videogame industry. In this paper, we describe existing evolutionary-based PCG systems and propose a universal method for videogame characters representation in such systems. An evolutionary PCG system based on this method was implemented as a part of this research. The system is universal and is well suited for integration in commercial and scientific game projects.

Keywords: procedural content generation, evolutionary computation, content personalization.

References

1. Ashlock D. Evolutionary Computation for Modeling and Optimization. New York: Springer, 2006. 571 p.
2. Ashlock D. Search-Based Procedural Generation of Maze-Like Levels // IEEE Transactions on Computational Intelligence and AI in Games. Sept. 2011. Vol. 3, No 3. P. 260–273.
3. Cook M. Multi-Faceted Evolution of Simple Arcade Games // IEEE Conference on Computational Intelligence and Games. Aug. 2011. P. 289–296.
4. Hendrikx M. Procedural Content Generation for Games: A Survey // ACM Transactions on Multimedia Computing, Communications, and Applications. Feb. 2013. Vol. 9, No 1. P. 1–22.
5. Hastings E.J. Automatic Content Generation in the Galactic Arms Race Video Game // IEEE Transactions on Computational Intelligence and AI in Games. Dec. 2009. Vol. 1, No 4. P. 245–263.
6. Martin A. Evolving 3D Buildings for the Prototype Video Game Subversion // Proceedings of the 2010 International Conference on Applications of Evolutionary Computation. Nov. 2010. Vol. 6024. P. 111–120.
7. Risi S. Combining Search-based Procedural Content Generation and Social Gaming in the Petalz Video Game // Proceedings of the 8th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. Oct. 2012. P. 63–68.
8. Shaker N. Evolving Levels for Super Mario Bros Using Grammatical Evolution // IEEE Conference on Computational Intelligence and Games. Sept. 2012. P. 304–311.
9. Sorenson N. A Generic Approach to Challenge Modeling for the Procedural Creation of Video Game Levels // IEEE Transactions on Computational Intelligence and AI in Games. Sept. 2011. Vol. 3, No 3. P. 229–244.

10. Stanley K.O. Real-Time Neuroevolution in the NERO Video Game // IEEE Transactions on Evolutionary Computation. Dec. 2005. Vol. 9, No 6. P. 653–668.
11. Togelius J. Search-Based Procedural Content Generation: A Taxonomy and Survey // IEEE Transactions on Computational Intelligence and AI in Games. Sept. 2011. Vol. 3, No 3. P. 172–186.
12. Yannakakis G.N. Experience-Driven Procedural Content Generation // IEEE Transactions on Affective Computing. July 2011. Vol. 2, No 3. P. 147–161.

Received 25 December 2013

ОПТИМИЗАЦИЯ МОДЕЛИРОВАНИЯ БЕЛКОВЫХ ВЗАИМОДЕЙСТВИЙ НА МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ С ПРЕДОСТАВЛЕНИЕМ ДОСТУПА К АЛГОРИТМУ ЧЕРЕЗ ВЕБ-ИНТЕРФЕЙС¹

К.В. Романенков, А.Н. Сальников

В работе представлены параллельные версии последовательной программы создания молекулярных интерфейсов с применением технологии OpenMP и MPI. Обе версии показали достаточную масштабируемость и лучшие временные показатели по сравнению с последовательной версией при запуске на одном процессоре. Моделирование белкового интерфейса для некоторых соединений занимает более двадцати часов счета на нескольких сотнях процессоров, поэтому для задач моделирования белковых соединений с большим количеством позиций важно наличие недетерминированных алгоритмов, позволяющих за приемлемое время получать биологически корректный результат. Выбор стохастических алгоритмов оправдал себя: и метод Монте-Карло, и алгоритм пчелиного поиска нашли пространственное расположение молекулы, соответствующее минимальному энергетическому состоянию. Предоставление доступа к реализации алгоритма по веб-интерфейсу отвечает современным тенденциям к перемещению вычислений на сторону сервера и позволяет широкому кругу специалистов использовать вычислительные мощности, предоставляемые Московским государственным университетом, а с учетом расширения сферы применимости задач молекулярного моделирования наличие открытого веб-интерфейса, предоставляющего удаленный доступ к вычислительным кластерам, является достаточно важной задачей.

Ключевые слова: биинформатика, многопроцессорные системы, создание молекулярных интерфейсов, параллельные алгоритмы, стохастические алгоритмы.

Введение

Существуют два основных подхода к моделированию белков: основанные на информации о пептидной последовательности и основанные на информации о структуре молекулы. Классическим представителем задачи первого типа является проблема фолдинга. Надо заметить, что на сегодняшний день подходы для решения задач этой категории далеко не очевидны, достаточно упомянуть, что рекордные длины искусственно сгенерированных белков с заданными свойствами трехмерной структуры составляют около трех сотен [1], при том, что, к примеру, в молекуле титина камбаловидной мышцы человека содержится более 25000 аминокислотных остатков.

Задача вычислительного моделирования белковых соединений относится ко второму типу задач. Одним из важнейших направлений в этой области считается создание молекулярных интерфейсов [2], позволяющее, в частности, предсказывать, какие аминокислотные остатки надо заменить в соединениях, не взаимодействующих в природе, но обладающих заданными свойствами, чтобы добиться их взаимодействия.

Существуют различные подходы для решения задач этого типа, из последовательных детерминированных версий можно выделить метод ветвей и границ [2] и алгоритм A* [3]. Общей проблемой этих методов являются большие временные затраты при обработке значительных объемов данных. В качестве методов, использующих распределенные вычисления,

¹Статья рекомендована к публикации программным комитетом Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: поиск новых решений – 2012».

можно выделить реализации, построенные на GRID-системах [4, 5], в которых заранее неизвестно, какое количество ресурсов каждый компьютер в сети сможет выделить для решения задачи, и реализацию, использующую технологию MPI для организации итеративного поиска решения [6], при этом большее число процессоров позволяет получать более точный результат и не ведет к ускорению работы.

Статья организована следующим образом. В разделе 1 описывается предметная область и обсуждаются особенности вычислительного моделирования белковых соединений. В разделе 2 содержится краткое описание используемых в исследовании алгоритмов и данных. Раздел 3 описывает модификацию исходной программы с целью созданию параллельных версий и использования стохастических алгоритмов. В разделе 4 описывается система *Aligner*, в которую была интегрирована созданная реализация. В разделе 5 приведены результаты вычислительных экспериментов и результаты работы стохастических алгоритмов. В заключении оценивается выигрыш от использования выбранных методов оптимизации и описываются направления будущих исследований.

1. Принципы компьютерного моделирования белковых соединений

Основной целью компьютерного моделирования белковых соединений является выбор аминокислот в воспроизводимой структуре, минимизирующих общую энергию системы [7]. Считается, что в каждом взаимодействующем домене (структурно обособленной единице) есть фиксированное число позиций, в которые возможно подставлять различные аминокислоты с целью получения минимального энергетического состояния (GMES – Global Minimum Energy Conformation). Гибкость аминокислот аппроксимируется ротамерами – конформационными изомерами, отличающимися от других конформеров углами поворота. Принята модель, в которой известно число углов вращения, принимающих конечное число значений. Данные о структурах ротамеров собраны в специальные библиотеки, где каждой совокупности значений углов вращения сопоставлен конформер. В рамках такой модели общая энергия системы складывается из трех основных компонент:

- 1) энергии остова, остающаяся неизменной при поиске GMES и поэтому не участвующая в оптимизации;
- 2) энергии взаимодействия ротамера с остовом молекулы;
- 3) энергии взаимодействия двух определенных ротамеров между собой.

Общая энергия вычисляется по формуле

$$\varepsilon(C) = E_1 + \sum_{i=1}^n E_2(C_i) + \sum_{i=1}^{n-1} \sum_{j=i+1}^n E_3(C_i, C_j), \quad (1)$$

где i – ротамер в i -й позиции.

Вычислительное моделирование белковых соединений имеет следующие ключевые особенности:

1. Количество позиций, участвующих в образовании интерфейса не очень велико (менее 100).
2. В каждой позиции моделируется мутация природного аминокислотного остатка, что влечет за собой рассмотрение большего количества ротамеров по сравнению с традиционными задачами структурного моделирования (синтез гомологов, при котором

рассматривается большее количество позиций, но с уже известными природными ротамерами, у которых необходимо рассмотреть лишь возможные пространственные изомеры).

3. Так как функция энергии допускает некоторую погрешность, в качестве результата работы алгоритма нужно получить список конформаций с минимальными энергиями, для которых производится честное моделирование, гораздо более требовательное к временным ресурсам.
4. Неоднородная энергетическая поверхность (это одна из сложностей применения алгоритмов самосогласованного поля (SCMF) [8]).
5. Проблема GMEC – NP-полная [8].

2. Параметры исследования

2.1. Описание используемых методов оптимизации

В качестве стохастических алгоритмов (то есть выдающих результат, зависящий не только от входных данных, но и от датчика случайных чисел), были выбраны MC/Q и SBC, так как они обладают достаточно потенциалом для выхода из локальным минимумов, которыми изобилует задача создания молекулярных интерфейсов. В качестве базового метода дискретной оптимизации использовался DEE.

DEE (Dead End Elimination). Простейший метод дискретной оптимизации, позволяет отсекаать заведомо неперспективные ротамеры, основываясь на их вкладе в общую энергию [8].

MC/Q (Monte Carlo / Quenching). Несмотря на то, что метод был разработан в середине 1950-х годов, успешно применяться в задачах моделирования белковых соединений он начал лишь в 1990-х годах [8]. Суть метода заключается в следующем: вначале последовательность ротамеров инициализируется случайным образом. Затем в случайной позиции происходит подстановка другого ротамера, причем замены на ротамеры разных аминокислот, включая ту, что стояла в выбранной позиции, равновероятны. После подстановки происходит вычисление энергии конформации, и если она меньше предыдущего значения, то замена принимается. Если новая энергия больше, то замещение подтверждается с вероятностью Больцмана, k – постоянная Больцмана. Вероятность Больцмана задает вероятность подстановки менее энергетически выгодного ротамера в определенную позицию и рассчитывается по формуле:

$$P = e^{\frac{-(E_{new} - E_{old})}{k \cdot T}} . \quad (2)$$

Величина T в данном случае исполняет роль температуры, позволяя избегать локальных минимумов. После завершения поиска возможен переход к фазе отжига. Для каждой позиции, выбранной в случайном порядке, перебираются все ротамеры аминокислотного остатка, который был найден в ходе работы метода Монте-Карло. Если энергия новой конформации будет меньше, то происходит замещение ротамера. Этот этап позволяет убедиться, что в полученной структуре отсутствуют подстановки отдельных конформеров, минимизирующих общую энергию. Отжиг имеет малую временную сложность, однако может серьезно улучшить найденное решение.

Алгоритм пчелиного поиска (Simulated Bee Colony). Алгоритм, появившийся в 2005 году, хорошо зарекомендовал себя как в задачах непрерывной, так и дискретной оптимизации [9]. Кратко его можно описать следующим образом: в начале работы случайным

образом выбирается m решений, каждое из которых представляет собой пчелу разведчика. Затем циклически лучшие n решений исследуются более тщательно: в зависимости от того является ли точка «элитной» или просто выбранной, в её окрестности исследуются s_e или s_p случайных решений, а остальные $(m - n)$ решений заменяются на случайные точки из пространства решений. Завершение алгоритма происходит либо в результате достижения определенной точности, либо после исчерпания числа итераций.

2.2. Описание входных данных для исследования взаимодействия структур

Для исследования были использованы два типа белковых структур, отражающих реальную задачу моделирования белковых интерфейсов. Постановка задачи для определенных белков сформирована институтом физико-химической биологии имени А.Н. Белозерского. Были исследованы: белковый комплекс LAGLIDADG эндонуклеаз и белковый комплекс антитело – антиген.

Эндонуклеазы — белки из семейства нуклеаз, узнающие длинные последовательности ДНК и вносящие в найденный фрагмент двунизовой разрыв. Они часто используются в генной инженерии для создания рекомбинантных ДНК, которые затем могут вводиться в клетки других организмов. Эндонуклеаза семейства LAGLIDADG состоит из двух доменов, и задача, связанная с ними, состояла в комбинировании отдельных доменов из различных белков с целью получения нуклеаз с новой специфичностью.

Антитела представляют собой специальные белки системы иммунитета. Они связываются с большой силой взаимодействия с антигеном (с высокой степенью диссоциации) — характерной частью патогена, например белковый капсид вируса, — и либо нейтрализуют антиген, либо маркируют его для других клеток иммунитета. Особенность строения антител заключается в том, что основной каркас молекулы не меняется, различаются только переменные петли, непосредственно отвечающие за взаимодействие с антигеном. Основным способом получения антител на сегодняшний день является иммунизация животных, это достаточно дорогая и неудобная процедура, поэтому становится актуальной задача оптимизации как фармакокинетических свойств антитела (растворимость, стабильность), так и его аффинности (прочность связи с антигеном). Можно выделить два направления развития в искусственном подборе антител:

- 1) оптимизация комплекса антитело – антиген средствами компьютерного моделирования для повышения аффинности антитела к антигену;
- 2) дизайн новых антител к указанным антигенам «de novo», имитирующий селективный отбор иммунной системы.

Относительно недавно были открыты и сейчас активно исследуются наноантитела [10, 11]: молекулы, сходные по структуре с антителами, но меньшего размера (в частности, меньше объем и количество переменных цепей), что дает им серьезные преимущества в синтезе лекарств перед антителами обычного размера. В качестве входных данных для задачи был предложен комплекс наноантитела с лизоцимом, который исполнял роль антигена. Стоит отметить вычислительную сложность этой задачи: исходя из того, что количество операций, необходимых для обработки 1 конформации, зависит от квадрата числа позиций, и зная общее количество комбинаций, получаемое перемножением числа всевозможных ротамеров в позициях, получаем, для моделирования наноантитела даже в 6 позициях (хотя может быть задействовано более 20) требует $6 \cdot 6 \cdot 191 \cdot 308 \cdot 190 \cdot 106 \cdot 196 \cdot 326$

операций с плавающей точкой, то есть требуется компьютер с производительностью более 2,7 ПФлопс.

3. Программа *FitProt*

Коллективом института им. А.Н. Белозерского в распоряжение авторов была предоставлена последовательная программа *FitProt* [12], написанная на языке программирования Python, которая осуществляла полный перебор конформаций для указанных пользователем позиций, предварительно применяя к ним фильтр DEE (см. пункт 2.1). На вход программе подаются два файла: со структурой соединения в формате PDB и с описанием в каких позициях и у каких ротамеров надо моделировать мутацию остатка. Второй файл имеет текстовый формат.

С целью ускорения работы последовательной реализации и для обеспечения возможности распараллеливания кода в *FitProt* были внесены некоторые изменения. Для хранения списка лучших конформаций, минимизирующих энергию системы, была реализована куча (heap), что было обусловлено меньшими временными затратами на добавление нового элемента и поддержания структуры кучи по сравнению с другими способами представления данных, а также скоростью сортировки. Выполнена реализация программы, использующая массив энергий, сгенерированный программой *FitProt* с использованием фильтра DEE, на языке C/C++, которая затем была распараллелена с использованием технологий OpenMP и MPI.

Авторами была произведена модификация программы *FitProt* так, чтобы ее было возможно запускать на многопроцессорных системах как с общей памятью, так и кластерной архитектуры. В OpenMP реализации все данные об энергиях ротамеров хранятся в общей памяти, а в MPI распределяются по процессорам. В обоих случаях параллельно рассматриваются различные конформации; затем из списков, сформированных различными MPI-процессорами, и OpenMP-нитьями, строится список минимальных энергий.

Для дальнейшего сокращения времени поиска решения ко входным данным были применены модифицированные алгоритмы MC/Q и GMES. Несмотря на то, что оба алгоритма предназначены для выдачи единственного решения, при создании реализации они были модифицированы для получения списка минимальных конформаций. Очевидно, что большое значение на результаты их работы оказывает точный выбор параметров, подходящих под решаемую задачу.

Алгоритм MC/Q реализован с применением технологии MPI, в котором между процессами распределяется пространство поиска, то есть большее количество процессоров не ведет к ускорению программы, зато теоретически может обеспечивать нахождение лучшего решения за то же время. Модификация алгоритма для получения списка минимальных конформаций состоит в следующем: согласно каноническому этапу MC на каждом процессоре ищется оптимальная конформация, после этого на этапе отжига при рассмотрении всех вариантов ротамеров аминокислотного остатка, стоящего в выбранной позиции, все полученные конформации добавляются в список решений. С увеличением количества позиций средняя энергия решения имеет тенденцию к снижению, что говорит о хорошем потенциале для масштабирования. Разделение пространства поиска между процессорами имеет целью оптимизацию найденных решений помимо GMES, когда на каком-то процессоре фиксируется локальный минимум и, возможно, улучшается на шаге отжига.

Реализован последовательный алгоритм пчелиного поиска, в качестве окрестности элитных точек рассматривались все варианты ротамеров аминокислотного остатка в случайно выбранной позиции, в качестве окрестности выбранных точек брались все ротамеры в случайной позиции с равной вероятностью. Количество элитных, выбранных и остальных точек соотносится как 10:25:100 соответственно, такие цифры, согласно эмпирическим исследованиям, позволяют получать оптимальный результат.

4. Описание системы Aligner

С целью обеспечения простоты использования создаваемого программного кода было решено интегрировать созданную параллельную версию в систему *Aligner* [13]. Изначально система *Aligner* создавалась как интернет-сервис для построения множественного выравнивания последовательностей на кластере и включала в себя систему авторизации, базу данных, возможность загрузки пользовательских данных, поддержку уведомлений о статусе заданий по электронной почте [14]. Преимущество *Aligner* над остальными веб-интерфейсами к многомашинным комплексам заключается в отсутствии у пользователя необходимости регистрации на вычислительных кластерах, к которым предоставляет доступ *Aligner*. Вместо этого достаточно завести учетную запись в системе, позволяющую ставить на счет задачи на всех суперкомпьютерах, связанных с *Aligner*. Структура системы *Aligner* представлена на рис. 1.

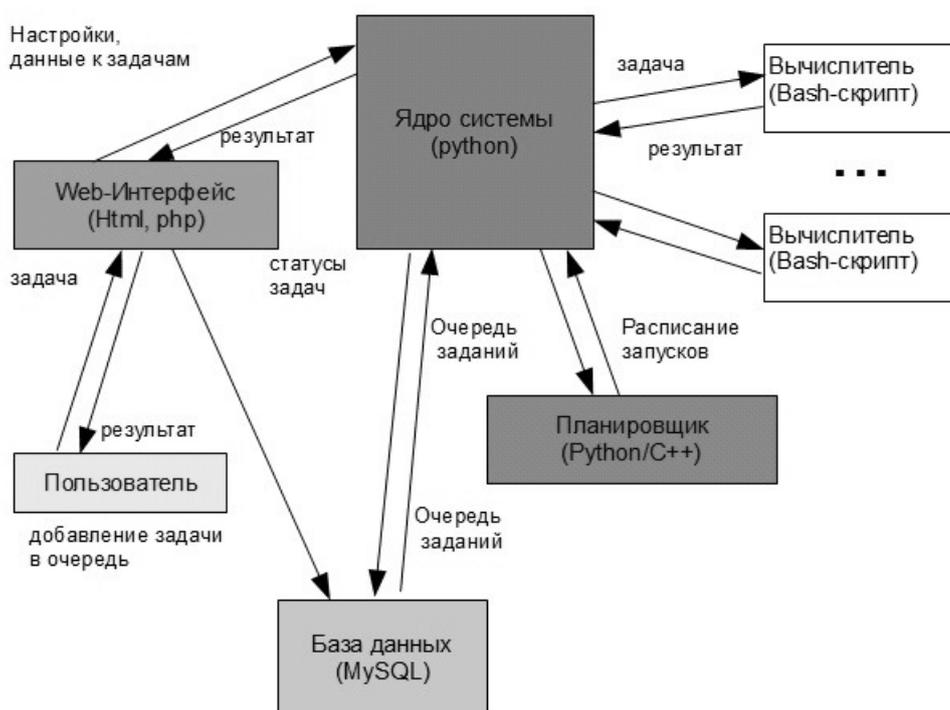


Рис. 1. Структура системы Aligner

В список алгоритмов, доступных для удаленного запуска на кластере, был добавлен FitProt. При выборе этого алгоритма, веб-интерфейс предлагает пользователю загрузить файл структуры в формате PDB и текстового файла, указывающего в каких позициях следует провести инжиниринг. После успешного считывания задания из базы данных происходит вызов исходного последовательного решения, которое с применением фильтра DEE

генерирует файл значений собственных и парных энергий, который отправляется на кластер, где ставится на счет программа, использующая его как параметр. В случае успешного завершения задания пользователю предоставляется возможность просмотреть и скачать выходные данные. Поведение программной системы в этом случае проиллюстрировано на рис. 2.



Рис. 2. Путь данных, при удаленном запуске заданий

5. Результаты

Для исследования эффективности параллельной реализации и как средство для удаленного запуска программы через веб-интерфейс была выбрана система Regatta, обладающая 16 процессорами Power4, располагающими 64 Гб общей памяти, и доступная из сети МГУ. Для расчетов молекулярного интерфейса наноантитела использовался комплекс «ЧЕБЫШЕВ», работающий на процессорах Intel Xeon E5472 3.0 ГГц и имеющий пиковую производительность 60 Тфлопс.

Табл. 1 содержит времена работы (в секундах) MPI версии программы, производившей моделирование молекулярного интерфейса для 3 позиций эндонуклеазы, и OpenMP версии программы, моделировавшей молекулярный интерфейс для 4 позиций эндонуклеазы. На рис. 3, приведенном ниже, указаны ускорения параллельных версий детерминированного алгоритма, исходя из данных табл. 1.

Во всех случаях стохастические алгоритмы нашли ГМЕС (конформация, соответствующая минимальной энергии, занимает первое место в списке, отсортированном по возрастанию энергии) и выдали несколько результатов из первой десятки минимальных конформаций. Лучшие результаты алгоритма пчелиного поиска объясняются способом получения списка оптимальной последовательности ротамеров: в MC/Q результат формируется в процессе отжига из лучшего найденного решения на этапе MC, когда перебираются конфор-

Таблица 1

Время работы параллельных версий программы (с)

Число процессоров	Время работы MPI версии	Время работы OpenMP версии
1	6,86	312
2	3,43	159
4	1,73	80
8	0,92	43
16	0,56	22

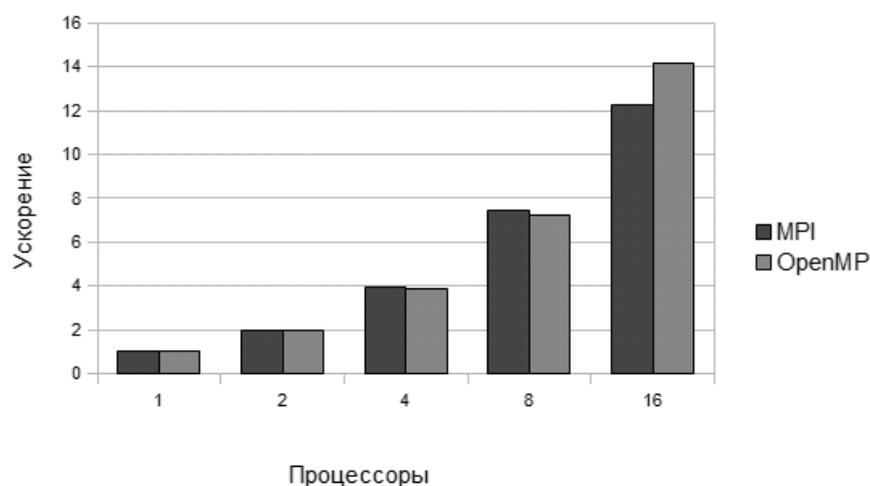


Рис. 3. Ускорение для параллельных версий детерминированного алгоритма

меры аминокислот, стоящих в решении, что ограничивает вариативность ответа. Алгоритм пчелиного поиска лишен этого недостатка, но существует опасность схождения всех решений к одному единственному глобальному минимуму при слишком долгом времени работы. В табл. 2 и табл. 3 приведены результаты работы алгоритмов, число найденных конформа-

Таблица 2

Результаты работы алгоритма MC/Q

Структура	Число позиций	Найдено конформаций
Эндонуклеазы	3	3
Эндонуклеазы	4	1
Эндонуклеазы, увеличенное число ротамеров	4	1
Наноантитело в комплексе с лизоцимом	4	4
Наноантитело в комплексе с лизоцимом	5	4

Таблица 3

Результаты работы алгоритма пчелиного поиска

Структура	Число позиций	Найдено конформаций
Эндонуклеазы	3	3
Эндонуклеазы	4	4
Эндонуклеазы, увеличенное число ротамеров	4	3
Наноантитело в комплексе с лизоцимом	4	3
Наноантитело в комплексе с лизоцимом	5	4

ций указано без учета ГМЕС, который был обнаружен во всех случаях. Число итераций при работе алгоритма пчелиного поиска с 3 позициями эндонуклеазы уменьшено на порядок, чтобы избежать схождения всех решений к ГМЕС.

Заключение

В работе были представлены параллельные версии последовательной программы создания молекулярных интерфейсов с применением технологии OpenMP и MPI. Обе версии показали достаточную масштабируемость и лучшие временные показатели по сравнению с последовательной версией. Выбор стохастических алгоритмов оправдал себя: и Монте-Карло, и алгоритм пчелиного поиска продемонстрировали высокую способность к выходу из локальных минимумов. Стоит упомянуть, что моделирование белкового интерфейса для наноантитела в 6 позициях занимает более 20 часов счета на нескольких сотнях процессоров, поэтому для задач моделирования белковых соединений с большим количеством позиций важно наличие недетерминированных алгоритмов, позволяющих за приемлемое время получать биологически корректный результат. Предоставление доступа к алгоритму по веб-интерфейсу отвечает современным тенденциям к перемещению вычислений на сторону сервера. Интеграция в систему *Aligner* позволяет широкому кругу специалистов использовать вычислительные мощности, предоставляемые Университетом, а с учетом расширения сферы применимости задач молекулярного моделирования, наличие открытого веб-интерфейса, предоставляющего удаленный доступ к вычислительным кластерам, является достаточно важной задачей.

В качестве направлений будущих исследований можно указать поиск оптимальных коэффициентов в функции энергии, например, с помощью методов машинного обучения и исследование применимости графических процессоров для ускорения работы программы.

Литература

1. Fortenberry, C. Exploring Symmetry as an Avenue to the Computational Design of Large Protein Domains / C. Fortenberry, E.A. Bowman, W. Proffitt, B. Dorr, S. Combs, J. Harp, L. Mizoue, J. Meiler // *Journal of the American Chemical Society*. — 2011. — Vol. 133, No. 45. — P. 18026–18029.

2. Wernisch, L. Automatic protein design with all atom force-fields by exact and heuristic optimization / L. Wernisch, S. Hery, S.J. Wodak // *Journal of Molecular Biology*. — 2000. — Vol. 301, No. 3. — P. 713–736.
3. Leach, A.R. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm / A.R. Leach, A.P. Lemon // *Proteins: Structure, Function, and Bioinformatics*. — 1998. — Vol. 33, No. 2. — P. 227–239.
4. Tantar, A.A. A parallel hybrid genetic algorithm for protein structure prediction on the computational grid / A.A. Tantar, N. Melab, E.G. Talbi, B. Parent, D. Horvath // *Future Generation Comp. Syst.* — 2009. — Vol. 23, No. 3. — P. 398–409.
5. Pitman, D.J. Improving computational efficiency and tractability of protein design using a piecemeal approach. A strategy for parallel and distributed protein design. — *Bioinformatics*. — 2013. / D.J. Pitman, C.D. Schenkelberg, Y.M. Huang, F.D. Teets, D. DiTursi, C. Bystroff. URL: <http://dx.doi.org/10.1093/bioinformatics/btt735> (дата обращения: 30.12.2013).
6. Molto, G. Protein design based on parallel dimensional reduction / G. Molto, M. Suarez, P. Tortosa, J.M. Alonso, V. Hernandez, A. Jaramillo // *Journal of Chemical Information and Modeling*. — 2009. — Vol. 49, No. 5. — P. 1261–1271.
7. Street, A.G. Computational protein design / A.G. Street, S.L. Mayo. URL: [http://dx.doi.org/10.1016/S0969-2126\(99\)80062-8](http://dx.doi.org/10.1016/S0969-2126(99)80062-8) (дата обращения: 24.02.2013).
8. Voigt, C.A. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design / C.A. Voigt, D.B. Gordon, S.L. Mayo // *Journal of Molecular Biology*. — 2000. — Vol. 299, No. 3. — P. 789–803.
9. Pham, D.T. The Bees Algorithm — A Novel Tool for Complex Optimisation Problems / D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi // *2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS'2006)* — 2006. — P. 454–461.
10. Hamers-Casterman, C. Naturally occurring antibodies devoid of light chains / C. Hamers-Casterman, T. Atarhouch, S. Muyldermans, G. Robinson, C. Hamers, E.B. Songa, N. Bendahman, R. Hamers // *Nature*. — 1993. — Vol. 363. — P. 446–448.
11. Revets, H. Nanobodies as novel agents for cancer therapy / H. Revets, P. De Baetselier, S. Muyldermans // *Expert Opinion on Biological Therapy*. — 2005. — Vol. 5, No. 1. — P. 111–124.
12. Grishin, A. Bioinformatics analysis of LAGLIDADG homing endonucleases for construction of enzymes with changed DNA recognition specificity / A. Grishin, I. Fonfara, W. Wende, D. Alexeyevsky, A. Alexeyevsky, S. Spirin, O. Zanegina, A. Karyagina // *4-th Moscow Conference on Computational Molecular Biology*. — Moscow, MSU, 2009. — P. 123.
13. Князев, Н. А. Система справедливого планирования и унифицированного запуска задач пользователя на суперкомпьютерах / Н.А. Князев, А.Н. Сальников // *Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа, 29 марта – 2 апреля 2010 г.)*. — Челябинск: Издательский центр ЮУрГУ, 2010. — С. 665–666.

14. Сальников, А.Н. Интернет-сервис для построения множественного выравнивания последовательностей на многопроцессорных системах, созданный на основе data-flow модификации алгоритма MUSCLE / А.Н. Сальников // Параллельные вычислительные технологии (ПАВТ'2009): Труды международной научной конференции (Нижний Новгород, 30 марта – 3 апреля 2009 г.). — Челябинск: издательский центр ЮурГУ, 2009. — С. 680–687.

Романенков Кирилл Владимирович, аспирант кафедры суперкомпьютеров и квантовой информатики факультета Вычислительной математики и кибернетики, Московский государственный университет (Москва, Российская Федерация), kromanenkov2@yandex.ru.

Сальников Алексей Николаевич, к.ф.-м.н., старший научный сотрудник факультета Вычислительной математики и кибернетики, Московский государственный университет (Москва, Российская Федерация), salnikov@cs.msu.ru.

Поступила в редакцию 22 ноября 2013 г.

*Bulletin of the South Ural State University
Series “Computational Mathematics and Software Engineering”
2014, vol. 3, no. 1, pp. 55–67*

OPTIMIZATION OF MODELING IMMOBILIZED PROTEIN INTERACTIONS ON COMPUTATIONAL CLUSTERS WITH THE SUPPLYING OF ACCESS TO THE ALGORITHM VIA WEB-INTERFACE

K.V. Romanenkov, Lomonosov Moscow State University (Moscow, Russian Federation),

A.N. Salnikov, Lomonosov Moscow State University (Moscow, Russian Federation)

In this work were introduced parallel versions based on OpenMP and MPI technologies of sequential program for modeling immobilized protein interactions. Both versions have shown good scalability and better time indices to compare with the sequential version when running on single processor. Need to mention that modeling of immobilized protein interactions for some compounds have taken more than twenty hours of computations on several hundreds of processors, that's why for such modeling tasks with the great quantity of positions availability of nondeterministic algorithms, providing biologically correct result in reasonable time, seems to be rather important. Selection of stochastic algorithms has proved its value: both Monte-Carlo and simulated bee colony algorithms had found conformation corresponding minimal energy state. Supplying of access to the algorithm via web-interface measures up modern specifications of remote computations and allows the wide circle of specialists use computational power of Moscow State University and, taking into account extending the sphere of application tasks of molecular simulation, the presence of open web-interface providing remote access to the computational clusters is quite an important task.

Keywords: bioinformatics, multiprocessor systems, immobilized protein interactions, parallel algorithms, stochastic algorithms

References

1. Fortenberry C., Bowman E.A., Proffitt W., Dorr B., Combs S., Harp J., Mizoue L., Meiler J. Exploring Symmetry as an Avenue to the Computational Design of Large Protein Domains // *Journal of the American Chemical Society*. 2011. Vol. 133, No. 45. P. 18026–18029.
2. Wernisch L., Hery S., Wodak S.J. Automatic protein design with all atom force-fields by exact and heuristic optimization // *Journal of Molecular Biology*. 2000. Vol. 301, No. 3. P. 713–736.
3. Leach A.R., Lemon A.P. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm // *Proteins: Structure, Function, and Bioinformatics*. 1998. Vol. 33, No. 2. P. 227–239.
4. Tantar A.A., Melab N., Talbi E.G., Parent B., Horvath D. A parallel hybrid genetic algorithm for protein structure prediction on the computational grid // *Future Generation Comp. Syst.* 2009. Vol. 23, No. 3. P. 398–409.
5. Pitman D.J., Schenkelberg C.D., Huang Y.M., Teets F.D., DiTursi D., Bystroff C. Improving computational efficiency and tractability of protein design using a piecemeal approach. A strategy for parallel and distributed protein design. *Bioinformatics*. 2013. URL: <http://dx.doi.org/10.1093/bioinformatics/btt735> (accessed: 30.12.2013).
6. Molto G., Suarez M., Tortosa P., Alonso J.M., Hernandez V., Jaramillo A. Protein design based on parallel dimensional reduction // *Journal of Chemical Information and Modeling*. 2009. Vol. 49, No. 5. P. 1261–1271.
7. Street A.G., Mayo S.L. Computational protein design. URL: [http://dx.doi.org/10.1016/S0969-2126\(99\)80062-8](http://dx.doi.org/10.1016/S0969-2126(99)80062-8) (accessed: 24.02.2013).
8. Voigt C.A., Gordon D.B., Mayo S.L. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design // *Journal of Molecular Biology*. 2000. Vol. 299, No. 3. P. 789–803.
9. Pham D.T., Ghanbarzdeh A., Koc E., Otri S., Rahim S., Zaidi M. The Bees Algorithm — A Novel Tool for Complex Optimisation Problems // *2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS'2006)*. 2006. P. 454–461.
10. Hamers-Casterman C., Atarhouch T., Muyldermans S., Robinson G., Hamers C., Songa E.B., Bendahman N., Hamers R. Naturally occurring antibodies devoid of light chains // *Nature*. 1993. Vol. 363. P. 446–448.
11. Revets H., De Baetselier P., Muyldermans S. Nanobodies as novel agents for cancer therapy // *Expert Opinion on Biological Therapy*. 2005. Vol. 5, No. 1. P. 111–124.
12. Grishin A., Fonfara I., Wende W., Alexeyevsky D., Alexeyevsky A., Spirin S., Zanagina O., Karyagina A. Bioinformatics analysis of LAGLIDADG homing endonucleases for construction of enzymes with changed DNA recognition specificity // *4-th Moscow Conference on Computational Molecular Biology. Moscow, MSU, 2009*. P. 123.
13. Knjazev N.A., Salnikov A.N. Sistema spravedlivogo planirovaniya i unificirovannogo zapuska zadach pol'zovatelja na superkomp'yuterah [Fair scheduling system for running user's tasks on supercomputers in unified mode]. *Parallel'nye vychislitel'nye tehnologii (PaVT'2010): Trudy mezhdunarodnoj nauchnoj konferencii (Ufa, 29 marta – 2 aprelya 2010) [Parallel Computational Technologies (PCT'2010): Proceedings of the International Scientific Conference (Ufa, Russia, March, 29 – April, 2, 2010)]*. Chelyabinsk, Publishing of the South Ural State University, 2010. P. 665–666.

14. A.N. Salnikov. Internet-cervis dlja postroenija mnozhestvennogo vyravnivaniya posledovatel'nostej na mnogoprocessornyh sistemah, sozdannyj na osnove data-flow modifikacii algoritma MUSCLE [Web-Service based on data-flow modification of MUSCLE algorithm for constructing multiple sequence alignment on multiprocessor systems]. Parallel'nye vychislitel'nye tehnologii (PaVT'2009): Trudy mezhdunarodnoj nauchnoj konferencii (Nizhni Novgorod, 30 marta – 3 aprelya 2009) [Parallel Computational Technologies (PCT'2009): Proceedings of the International Scientific Conference (Nizhni Novgorod, Russia, March, 30 – April, 3, 2009)]. Chelyabinsk, Publishing of the South Ural State University, 2009. P. 680–687.

Received 22 November 2013

О НЕКОТОРЫХ ПРОБЛЕМАХ «ЗАОБЛАЧНОГО» МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ¹

В.П. Ильин

Разработка интегрированного прикладного программного окружения для математического моделирования, ориентированного на решение междисциплинарных прямых и обратных задач широким кругом пользователей в рамках облачных вычислений на суперцентрах петафлопсного уровня ставит большой круг вопросов методического, технологического и организационного характера. В работе рассматривается концепция базовой системы моделирования (БСМ), функционирование которой предполагается в рамках вычислительного центра коллективного пользования (ВЦКП). БСМ представляет собой интегрированное программное окружение, поддерживающее все основные технологические этапы крупномасштабных вычислительных экспериментов, адаптируемое к эволюции моделей и алгоритмов решения задач, а также архитектур многопроцессорных вычислительных систем (МВС) и ориентированное на длительный жизненный цикл, обеспечиваемый эффективной генерацией пакетов прикладных программ (ППП) для конкретных предметных областей. Проблема достижения высокой суммарной производительности ВЦКП рассматривается как задача условной оптимизации расписания потока расчетных заданий с учетом используемых ресурсов конкретных приложений.

Ключевые слова: масштабируемый параллелизм, облачные вычисления, математическое моделирование, многопроцессорная вычислительная система, базовая система моделирования, прикладное программное обеспечение.

Введение

Одна из актуальных проблем вычислительного сообщества заключается в кардинальном повышении востребованности компьютерных расчетов как третьего пути познания процессов и явлений, наряду с теоретическими исследованиями и натурными экспериментами. Другими словами, математизация различных наук и моделирование должны сыграть значительную инновационную роль в многочисленных производственных и социальных сферах, сравнимую по масштабам с информационными потрясениями, привнесенными массовым внедрением мобильных телефонов и интернета в течение последних десятилетий.

С точки зрения математического моделирования, требующего выполнения крупномасштабных вычислительных экспериментов при решении прямых и обратных междисциплинарных задач, ориентированных на достижение качественно нового уровня получения принципиальных фундаментальных знаний и производственных решений, в суперкомпьютерном развитии следует отметить две главные тенденции. С одной стороны — это концентрация экстремальных ресурсов в рамках центров коллективного пользования (Data Centers, [1]) пост-петафлопсных масштабов с миллионами разнородных компонент, включая многоядерные центральные универсальные и специализированные ускорительные процессорные устройства, программируемые логические интегральные схемы (ПЛИС), а также вычислительные сети (Grid), объединяющие удаленные расчетные мощности.

Другая тенденция — это формирование математического и программного обеспечения в рамках многопользовательской вычислительной интегрированной среды, поддерживающей основные технологические этапы реализации наукоемких проблем: геометрическое и

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

функциональное моделирование, генерация сеток, аппроксимация и дискретизация исходных уравнений, решение алгебраических задач, оптимизационные подходы к решению обратных задач, пост-обработка и визуализация результатов, средства принятия решений по итогам расчетов, см. [2–4]. В некотором смысле примерами программных прототипов могут служить такие комплексы, как OpenFOAM [5] и DUNE (Distributed Unified Numerical Environment [6]), или более узко направленные OpenCascade [7] и Salome [8] .

Мотивации такого пути развития заключаются в следующем:

- наличие общих эволюционных направлений создания открытых программных систем (Open Source), в соответствии с разработанной международным комитетом под руководством Дж.Донгарры «дорожной карты» экзавычислений [9], а также принципов открытых инноваций, см. [10, 11];
- несоответствие современным условиям суперкомпьютерного развития широко распространенных традиционных проблемно-ориентированных пакетов прикладных программ (ППП) типа ANSYS [12], NASTRAN [13] и многих других, имеющих историю развития еще с прошлого века, а также узкая область применимости различных алгоритмических библиотек (алгебраических, графических и т.д.);
- насущная потребность создания вычислительных интегрируемых программных окружений с длительным жизненным циклом, адаптируемых к эволюции компьютерных платформ, легко развиваемым и решающим проблему взаимодействия с внешними разработками, а также скрывающим от конечных пользователей излишние технические детали путем предоставления дружественных интеллектуальных интерфейсов; фактически здесь идет речь о резком повышении интеллектуальности прикладного программного обеспечения (ППО), связанной с автоматизацией построения математических моделей и алгоритмов или, образно говоря, о переходе от палео-информатики к нео-информатике [14].

Целью разработки таких инструментальных сред является создание предпосылок для инновационного внедрения и массового применения математического моделирования путем оперативной генерации приложений нового поколения для фундаментальных исследований или анализа и оптимизации производственных процессов. Необходимо иметь в виду, что темпы развития современных и будущих высокотехнологичных производств (новые материалы, био- и нано-технологии, и пр.) напрямую зависят от уровня математизации и компьютеризации данных сфер деятельности, а главным орудием познания здесь становится моделирование. Следует даже подчеркнуть, что компьютерное моделирование наиболее актуально именно в прорывных технологиях, где традиционные подходы исследований имеют ограниченную применимость. Другая важная диалектическая тенденция заключается в том, что наряду с активным развитием различных научных дисциплин, в последние десятилетия идет бурное накопление результатов в теоретической, прикладной и вычислительной математике, которые ждут своего компьютерного воплощения. А это означает, что новое ППО должно быть «живым организмом», комфортная перманентная эволюция которого требует самой серьезной поддержки.

В данной работе излагается концепция коллективной (многопользовательской) базовой системы моделирования (БСМ), включая архитектуру построения разработки и особенностей ее функционирования в рамках облачных вычислений и технологий программно-вычислительных услуг (SaaS – Software as a Service). В п. 1 рассматриваются вопросы стратегического и тактического планирования параллельных вычислений с точки зрения

различных целевых функций в оценках эффективности использования технического оборудования вычислительного центра коллективного пользования (ВЦКП). Надо сказать, что новое — это хорошо забытое старое, и в определенной степени аналогичные проблемы обсуждались в работе [15] 1988 года. Пункт 2 посвящен некоторым системным аспектам, возникающим при реализации предлагаемых приложений, а также их возможным обобщениям и дальнейшим путям развития.

1. Технологии и критерии эффективности распараллеливания

Одной из главных характеристик параллелизма вычислительного процесса или алгоритма (или же решаемой задачи) является масштабируемость, которая, вообще говоря, может пониматься по-разному. Сама задача, с точки зрения ее вычислительной сложности, или ресурсоемкости, специфицируется количеством необходимых арифметических операций C и памяти M , которые естественно измерять во флопах (действия с плавающей запятой) и машинных словах, причем в современном моделировании следует понимать стандартные 64-разрядные представления чисел, достаточно надежно гарантирующие требуемую на практике точность. В свою очередь, величины C и M зависят от количества степеней свободы N , которое может определяться, например, числом узлов сетки или порядком системы линейных алгебраических уравнений (СЛАУ).

Под масштабируемостью можно понимать в простейшем случае ускорение расчетов, пропорциональное увеличению количества процессоров P . Несколько отличное от этого понятие — сохранение времени решения задачи при одновременном увеличении ее ресурсоемкости и количества процессоров в одинаковое число раз. Вообще говоря, зачастую в распараллеливании главное — это возможность интегрирования памяти для решения большой задачи, которую при малом количестве процессоров просто нельзя реализовать.

Для некоторого абстрактного рассмотрения многоцелевого использования коллективных компьютерных ресурсов необходимо в первую очередь выбрать формальную модель многопользовательской вычислительной системы, а также и самого вычислительного процесса.

В достаточно общем виде мы можем рассмотреть неоднородную, или гетерогенную, многопроцессорную — многоядерную вычислительную систему (или сеть), которую будем сокращенно называть МВС, как совокупность соединенных между собой различных кластеров и серверов. Кластер, в свою очередь, представляется набором многопроцессорных узлов, в которых процессоры могут быть или «стандартные» (центральные, CPU), содержащие относительно небольшое количество вычислительных ядер (например, 8 или 12) с общей памятью, или специальные (наиболее распространенные — это графические процессоры общего назначения, GPGPU), насчитывающие несколько сот ядер (например, 512). Представляемая для расчетов память имеет, как правило, иерархическую структуру и включает оперативную память процессоров и три уровня кэша, отличающиеся между собой объемом и быстродействием. Отдельный сервер формально можно считать кластером, состоящим из одного «большого» узла, содержащего достаточно емкую общую оперативную память и значительное количество многоядерных процессоров различных типов.

Вычислительный сеанс на такой МВС заключается в управляемом диспетчером выполнении арифметических действий на соответствующих устройствах, а также в информационных обменах между ними. Идеальным является выполнение расчетов непосредственно в

процессоре с использованием только сверхбыстрых регистров памяти, однако их количество ограничивается несколькими десятками.

Ограничиваясь тривиальной моделью вычислений, время $t(A)$ реализации фрагмента алгоритма (или задачи) A на каком-то одном процессоре компьютера можно выразить формулой

$$t(A) = t_a + t_c + t_w, \quad t_a = \tau_a N_a, \quad t_c = (\tau_0 + \tau_c N_c) m_c, \quad (1)$$

где τ_a и N_a – среднее время выполнения одного действия и общее количество арифметических операций, t_c – время реализации межпроцессорных информационных обменов, τ_c – время межпроцессорной передачи одного числа, N_c – количество передаваемых чисел, τ_0 – время задержки (настройки) одной транзакции, m_c – число обменов, а t_w – время простоев (ожидания) процессора, которые можно рассматривать как неизбежную плату за рассогласованность структуры конкретного алгоритма и архитектуры МВС, на которой он выполняется. Для реальных вычислительных систем имеют место соотношения $\tau_a \ll \tau_c \ll \tau_0$, в силу чего повышение производительности расчетов в значительной степени связано с минимизацией коммуникаций. Здесь следует отметить и другой немаловажный фактор: операции передачи данных являются намного более энергозатратными, в сравнении с арифметическими действиями, что сильно влияет на стоимость эксплуатации МВС (потребление электроэнергии суперкомпьютера постпетафлопсной производительности достигает нескольких мегаватт).

Несовершенство формул в (1) с введением величины τ_a сродни понятию «средней температуры по больнице», поскольку реальная длительность различных арифметических действий сильно отличается друг от друга и, более того, на самом деле необходимо учитывать совокупную скорость выполнения арифметических выражений на процессорном конвейере.

Реально единицей измерения времени в компьютере является «такт», длительность которого определяется частотой электропитания и составляет несколько гигагерц. Длительность выполнения каждого арифметического действия характеризуется количеством тактов и, например, при «хорошей» организации вычислений выражения $y_i = ax_i + b, i = 1, \dots$, значение каждого y_i будет занимать только один такт.

Второе грубое приближение заключается в том, что τ_0 и τ_c тоже представляют собой усредненные характеристики скоростей обменов между различными устройствами. В пределах одного CPU — это связи умножителей и сумматоров с процессорными регистрами, обмены между регистрами и кэшем, а также между различными уровнями оперативной общей памяти. Намного медленнее идет передача данных между CPU и GPU, между вычислительными узлами одного кластера и тем более — между различными кластерами, соединяемыми в рамках ВЦКП с помощью Grid-технологий. В последнем случае итоговое быстроедействие определяется техническими характеристиками телекоммуникационных каналов.

С другой стороны, значительного уменьшения расчетного времени $t(A)$ можно добиться путем специальной организации вычислительного процесса: совмещение обменов и арифметических операций во времени, организация информационных буферов, различные приемы оптимизации программного кода типа «развертка циклов», и т.д.

Качество распараллеливания задачи A на P процессорах в первом приближении можно охарактеризовать двумя параметрами — коэффициентом ускорения S_P и эффективностью

использования процессоров:

$$S_P(A) = \frac{t_1(A)}{t_P(A)}, \quad E_P(A) = S_P(A)/P, \quad (2)$$

где $t_P(A)$ – время выполнения задачи на P процессорах. Естественно, в данном случае процессор подразумевается одноядерным, чего на практике уже почти не бывает. Если же в (2) за P принимать количество ядер, то здесь надо учитывать возможные конфликты при работе с общей памятью. В это ситуации надо иметь в виду, что оцениваемая теоретических пиковая производительность может сильно отличаться от реальной.

В определенном смысле идеальным считается линейное распараллеливание, когда ускорение S_P пропорционально числу процессоров P , а эффективность E_P близка к единице. На практике зачастую считаются приемлемыми значения $E_P = 0.1 \div 0.5$. Однако иногда случаются и чудеса со значениями $E_P > 1$ (сверхлинейное ускорение), когда, например, с удвоением P алгоритм, изначально выполняемый в медленной памяти, начинает «умещаться» в кэше.

Здесь и ниже под числом понимается вещественная переменная со стандартной двойной точностью в 64-битовом представлении с плавающей запятой, что является необходимым (и зачастую достаточным) при решении больших задач, например — при решении СЛАУ с размерностью порядка миллиона. Отметим, что в памяти объемом один гигабайт размещается $128 \cdot 10^6$ таких чисел. Отсюда, например, следует, что для хранения элементов разреженной квадратной матрицы с размерностью $N = 10^9$ и со средним числом ненулевых элементов в строке, равным 100, требуется около одного терабайта памяти. Это означает, во-первых, что размещение такого объема памяти требует большого числа процессоров, а во-вторых, что соответствующие межпроцессорные пересылки представляют весьма дорогую операцию.

Необходимо заметить, что ресурсоемкость вычислительных задач и компьютеров эволюционирует при сохранении баланса между быстродействием и объемом памяти МВС: терафлопсам соответствуют терабайты, петафлопсам – петабайты и т.д. Такая закономерность приводит к тому, что реализация большой расчетной задачи на МВС — это фактически комплексная вычислительно-информационная проблема (Intensive Data Computing). Хотя стоит подчеркнуть, что чисто информационные задачи традиционно относятся не к математическому моделированию, а к другой проблематике — Big Data или Data Mining.

Парадоксальным, на первый взгляд, является тот факт, что аналитические оценки компьютерной производительности являются «терра инкогнито» вычислительной информатики. Реальные операции при наличии конвейеров, многоуровневой памяти и внутренних особенностях компиляторов настолько сложны, что даже смоделировать выполнение простейших векторно-матричных выражений не представляется возможным. Поэтому проблема оптимизации программного кода приоткрыта только немногим посвященным, а единственный реальный путь ее исследования и понимания — это систематический эксперимент с квалифицированными измерениями, в которых должны помогать инструментальные профилировщики вычислительных процессов. Справедливости ради надо сказать, что многие частные приемы распараллеливания поддерживаются достаточно профессионально или компиляторами, или библиотечными функциями, что зачастую освобождает от излишней головной боли прикладных математиков и программистов.

Отметим, что попытки построения систем автоматического распараллеливания алгоритмов (типа долго развиваемого, но затем закрытого, проекта HPF — High Performance Fortran) пока не завершились созданием промышленного компилятора. Существующие же

средства распараллеливания — различные варианты системы передачи сообщений MPI для распределенной по кластерным узлам памяти и инструменты OpenMP или CUDA для общей памяти многоядерных CPU или GPU имеют «полуручной» принцип управления. При этом классическая проблема «отображение алгоритмов на архитектуру ЭВМ» фактически сводится к интуитивному сопоставлению MPI-процессов и вычислительных потоков конкретному доступному составу процессоров. Вопросы же высокопроизводительных вычислений в «облаках» и в сетевых технологиях в большой степени относятся не непосредственно к эффективности распараллеливания алгоритмов, а к управлению расчетными заданиями и к интегрированию компьютерных ресурсов, см. [16].

В заключение данного пункта укажем, что в разные годы изобреталось большое количество специализированных вычислительных устройств, которые для определенного класса задач или алгоритмов достигали рекордных скоростей. Однако эти спецпроцессоры не выдержали рыночной конкуренции с универсальными МВС. Возможным исключением являются реконфигурируемые ПЛИС [17], однако вопросы высокопроизводительных вычислений с их использованием представляют собой самостоятельную проблему, выходящую за рамки данной статьи. В идеале представляется процесс конвергенции между создателями компьютерных архитектур и разработчиками математического и программного обеспечения, результатом чего было бы осуществление голубой мечты и математиков, и конструкторов – оперативное проектирование сверхбыстрых устройств «под задачу» или «под алгоритм».

2. О концепции «заоблачного» моделирования

Для качественного ППО, которое могло бы способствовать инновационности, массовой востребованности и практической эффективности математического моделирования на современных МВС, можно сформулировать следующие технические требования:

- применимость к решению широкого класса проблем: прямые и обратные междисциплинарные задачи, описываемые системами дифференциальных и интегральных уравнений в многомерных расчетных областях со сложной геометрией и контрастными материальными свойствами, включая стационарные и динамические, линейные и нелинейные постановки;
- высокое разрешение численных решений, определяемое адекватностью математических моделей и аппроксимационными качествами методов, адаптируемых под свойства задачи, а также робастностью (надежностью и безотказностью) алгоритмов;
- высокая производительность расчетов, включая масштабируемую параллельность с отображением алгоритмов на архитектуру гетерогенных многопроцессорных и многоядерных вычислительных систем;
- интеллектуальность программных реализаций, обеспечивающая автоматическое формирование моделей и построение алгоритмов, управление вычислительным процессом и создание дружественных профессиональных интерфейсов для различных категорий пользователей;
- гибкие конфигурационные свойства, способствующие длительному жизненному циклу ППО с непрерывным пополнением состава применяемых моделей и алгоритмов, а также адаптации к изменениям компьютерных платформ;
- открытость к внешним программным разработкам и возможность переиспользования сторонних продуктов.

Перечисленные экстремальные условия ориентированы главным образом на решение суперзадач на суперкомпьютерах постпетафлопсного уровня, а их воплощение предполагает создание проекта не отдельной группы разработчиков, а сообщества разнопрофильных специалистов с широкой координацией и кооперацией многолетнего фронта работ.

Реализация данного амбициозного круга проблем рассматривается в рамках базовой системы моделирования, структура которой включает следующие основные части:

- интегрированное инструментальное ядро, состав которого определяет класс решаемых задач и применяемых алгоритмов;
- конфигурационные средства, ответственные за уровень интеллектуальности и пользовательской доступности разрабатываемого ППО;
- семейство ППП и алгоритмических библиотек, характеризующих конечную практическую направленность и эффективность создаваемых приложений.

Ядро БСМ представляет собой методо-ориентированное окружение, предназначенное для поддержки всех основных технологических этапов крупномасштабного вычислительного эксперимента, а его функциональное наполнение определяется следующим набором компонент, или подсистем:

- геометрическое и функциональное моделирование, на основе которого реализуется автоматизация когнитивных методик, формируются профессиональные пользовательские интерфейсы для конкретных приложений, а также импорт-экспорт с САПРами и внешними графическими пакетами;
- генерацию адаптивных неструктурированных сеток, с поддержкой параллельных методов декомпозиции сеточных подобластей, многосеточных подходов и локальных сгущений дискретизации в окрестности сингулярностей решений, на основе априорного и/или апостериорного анализа решаемой задачи;
- сеточные и спектральные алгоритмы различных порядков аппроксимации исходных задач, обеспечивающие высокое разрешение численных решений и апостериорные оценки точности (методы конечных объемов, конечных элементов, методы граничных интегральных уравнений, разрывные схемы Галеркина и др.);
- параллельные алгебраические решатели для возникающих в практических задачах больших плохо обусловленных систем сеточных линейных и нелинейных уравнений, а также для решения проблем собственных значений соответствующих матриц;
- методы оптимизации для решения проблем условной минимизации, локальной или глобальной, как основной подход к реализации обратных задач (идентификация параметров модели, автоматизация проектирования, оптимизация эксплуатационных режимов и т.д.);
- обработка и визуализация результатов численных расчетов;
- управление вычислительным процессом, формирование и реализация сценариев моделирования;
- анализ результатов моделирования и средства принятия решений.

Важно подчеркнуть, что каждая из подсистем ядра является открытой и легко пополняемой инструментальной средой для соответствующей технологической стадии, которая практически не зависит от других компонент, а взаимодействие между ними происходит через унифицированные структуры данных: геометрические (ГСД), функциональные (ФСД), сеточные (ССД), алгебраические (АСД) и т.д., – со множественными представлениями и возможностями конвертизации, для обеспечения гибких внутренних интерфейсов между

различными вариантами методов или моделей и внешних контактов со сторонними программными продуктами. Подчеркнем, что многоверсионность данных и алгоритмов требует специальных средств поддержки, но зато позволяет достичь такого уникального качества, как совмещение универсальности и эффективности разработки, за счет ее адаптивности к специфическим свойствам задачи.

Помимо перечисленных алгоритмических компонент, функциональное наполнение должно включать средства методической поддержки математического и программного обеспечения: верификации, тестирования, сравнительного анализа, т.е. экспериментального исследования. Это подразумевает создание каталогизированной библиотеки тестовых примеров и типовых задач с архивами расчетных результатов и соответствующими описаниями.

Средства конфигурационного управления составляют системное наполнение БСМ и обеспечивают автоматизированное пополнение состава математических моделей решаемых задач и применяемых алгоритмов, конструирование вычислительных схем в составе ППП для классов приложений, диалоговых графических интерфейсов для профессионалов в прикладных областях, а также адаптацию на конкретные компьютерные платформы и операционные обстановки. Фактически набор системных средств определяет интеллектуальный уровень БСМ, а их реализация предполагает активное использование специализированных языков программирования [14]. Главное назначение системного блока — это поддержка длительного жизненного цикла всего проекта, который должен обеспечиваться пользователями высокого уровня — разработчиками новых алгоритмов и приложений в составе БСМ.

Семейство ППП, образуемое конфигуратором из инструментальных модулей ядра, представляет собой непосредственное орудие математического моделирования и определяет производственные сферы его реального эффективного применения.

В целом конгломерат предлагаемых в составе БСМ приложений — это и есть непосредственный пакет решений для конечного пользователя, которому неинтересны (а в определенном смысле — даже вредны) знания о «внутренней кухне» информационно-вычислительных технологий, а нужны реальные возможности продукта для исследования его конкретной проблемы. В некотором смысле идеальным является представление ППП в виде «черного ящика» с минимальным набором кнопок и инструкций. Однако на практике неизбежны и компромиссные варианты в форме «серых ящичков» с возможностями модификаций режимов моделирования для заинтересованных продвинутых конечных пользователей.

Типовая вычислительная блок-схема для решения стационарной линейной краевой задачи представляется следующим образом: формирование с помощью графического редактора исходных данных о расчетной области и решаемых уравнениях (в виде ГСД и ФСД), построение сетки (ССД), реализация аппроксимаций и формирование АСД, решение СЛАУ, пост-обработка, визуализация и анализ полученных результатов. Если поставленная проблема является нелинейной, то необходимые этапы (а возможно, и все) повторяются с проведением итераций по нелинейным членам. В случае нестационарности моделируемого процесса рассмотренные этапы выполняются последовательно для различных шагов по времени. И наконец, для обратных задач с формулируемым целевым функционалом, зависящим от некоторых параметров в исходных данных, все предыдущие стадии, представляющие собой решение одной прямой задачи, циклически повторяются в соответствии с оптимизационными алгоритмами условной минимизации функционала.

Таким образом, формально вычислительная схема решения одной абстрактной большой задачи в общем случае представляется вложенными циклами, в каждом из которых распараллеливание алгоритмов может осуществляться в соответствии со своими тактиками и стратегиями.

Начать рассмотрение следует с нижнего уровня — этапа решения СЛАУ, который играет ключевую роль в силу того, что его ресурсоемкость нелинейным образом растет с увеличением порядка системы, или числа степеней свободы. Главным орудием распараллеливания здесь является метод декомпозиции расчетной области на подобласти (с пересечениями или без), в каждой из которых независимо и синхронно (в идеальном случае) решаются соответствующие подсистемы, а с их помощью устраивается итерационный процесс по подобластям.

Одно из важных условий распараллеливания методов решения современных сверхбольших СЛАУ на МВС — это отсутствие программных ограничений на порядки решаемых систем и на количество используемых процессоров и/или ядер. По этой причине реализацию алгоритмов надо делать в строго распределенном варианте, без чрезмерной концентрации ресурсов на корневом процессе системы MPI.

Отметим еще один серьезный момент: во избежание простоя вычислительного оборудования при одновременном решении вспомогательных задач в подобластях разными MPI-процессами декомпозиция расчетной области должна быть сделана сбалансированной. Последнее понятие, строго говоря, означает, что алгебраические подсистемы для подобластей должны решаться за одинаковое время, которое в первом приближении определяется порядком, т.е. количеством строк в соответствующем блоке глобальной матрицы.

Разбиение краевой задачи на подзадачи может рассматриваться в трех ипостасях: геометрическом, сеточном и алгебраическом. Наиболее близки между собой две последние интерпретации, поскольку каждому узлу сетки соответствует своя строка матрицы, а ребру сеточного графа соответствуют симметрично расположенные ненулевые внедиагональные элементы матрицы.

Алгебраическая декомпозиция предполагает, что дана только информация о портрете большой разреженной матрицы порядка N , представленной в одном из сжатых форматов, например — CSR (Compressed Sparse Row), когда хранятся только ненулевые элементы матрицы и информация об их расположении. Задача в данном случае сводится к формированию CSR-форматов для заданного числа блочных строк $P \ll N$, причем в одном блоке должны быть собраны наиболее тесно связанные между собой матричные строки. Проблема эта решается методами преобразования графов и является довольно ресурсоемкой.

Более перспективным является сеточная декомпозиция на этапе дискретизации расчетной области. Сеточная структура данных является намного более экономичной, чем АСД, поэтому достаточно большую сеточную расчетную область можно создать в корневом MPI-процессе, а затем из нее сформулировать сеточные подобласти в распределенном по процессорам варианте. Более того, после генерации исходных сеток в подобластях к ним можно применить несколько раз процедуру сгущения и получить семейства иерархических вложенных сеток. После окончания этапа дискретизации выполнение этапа аппроксимации и формирования АСД для подобластей проводится сразу в параллельной (распределенной) версии, без чрезмерной перегрузки корневого MPI-процесса.

Отметим, что проблема декомпозиции на самом верхнем — геометрическом — уровне с информационной точки зрения вообще не представляет трудностей, поскольку объемы

данных в ГСД и ФСД малы, так что они легко могут быть растиражированы по всем процессам.

Рассмотренная выше алгоритмическая структура может составлять часть схемы расчетов серии однотипных задач, которые должны выполняться или одновременно, или только последовательно. Такие вычислительные сеансы с решениями однотипных прикладных проблем могут быть как эпизодические, так и составлять регулярную область деятельности некоторой группы пользователей. В таких случаях целесообразно формирование некоторого «частного облака» с выделением виртуального кластера проблемно-ориентированной конфигурации, производительность которого может быть оптимизирована на основе анализа соответствующего вычислительного процесса.

Следующий уровень организации вычислений в абстрактном плане — это решение не связанных между собою задач независимыми пользователями, объединяемых только общностью используемого прикладного программного обеспечения, т.е. БСМ. Здесь проблемы управления интегрированными ресурсами ВЦКП и планирования компьютерных экспериментов требуют создания некоторой инфраструктуры с использованием не только облачных, но и сетевых технологий. Соответствующие решения требуют еще самих определений и оптимизационных постановок, понимание и актуальность которых неизбежно приходит с погружением в практику массового математического моделирования.

Заключение

Главный вывод данной работы состоит в том, что математическое моделирование представляет актуальнейшую инновационную проблему для всевозможных сфер человеческой деятельности, и для ее решения имеются все объективные предпосылки в достижениях суперкомпьютеров, информационных технологий и математических методов. Однако воплощение видимых здесь перспектив в реальность представляет собой самостоятельную задачу поиска новых прорывных решений, в которых положительный результат невозможен без интегрированных целенаправленных усилий разноплановых профессионалов вычислительного сообщества; компьютерщиков, программистов, математиков и представителей конкретных прикладных областей.

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

Литература

1. DATA CENTER: URL: www.datacenterknowledge.com (дата обращения: 01.12.2013).
2. Ильин, В.П. Технологии вычислительного программирования / В.П. Ильин, И.Н.Скопин // Программирование. — 2011. — № 4. — С. 53–72.
3. Голубева, Л.А. О программных технологиях в геометрических аспектах математического моделирования / Л.А. Голубева, В.П. Ильин, А.Н. Козырев // Вестник НГУ, серия «Информационные технологии». — 2012. — Т. 10. — № 2. — С. 25–33.
4. Ильин В.П. DELAUNAY: технологическая среда генерации сеток / В.П. Ильин // СибЖИМ. — 2013. — Т. 16, — № 2(54). — С. 83–97.
5. OpenFOAM® — The Open Source Computational Fluid Dynamics (CFD) Toolbox: URL: www.openfoam.com (дата обращения: 01.12.2013).

6. DUNE Numerics: URL: www.dune-project.org (дата обращения: 01.12.2013).
7. OPEN CASCADE: URL: www.opencascade.org (дата обращения: 01.12.2013).
8. SALOME: URL: www.salome-platform.org (дата обращения: 01.12.2013).
9. IESP: URL: www.exascale.org/iesp (дата обращения: 01.12.2013).
10. Чесбро, Г. Открытые инновации / Г. Чесбро. — М., изд. «Поколение». — 2007. — 333 с.
11. OPEN NOVATION: URL: www.opennovation.org (дата обращения: 01.12.2013).
12. ANSYS — Simulation Driven Product Development: URL: www.ansys.com (дата обращения: 01.12.2013).
13. MSC Nastran — Industry Leading Multidisciplinary FEA: URL: www.mssoftware.com/product/msc-nastran (дата обращения: 01.12.2013).
14. Kleppe, A. Software Language Engineering: Creating Domain-Specific Language Using Metamodels / A. Kleppe // N.Y. Addison-Wesley. — 2008. — 207 p.
15. Алексеев, А.С. Комплексный центр математического моделирования: концепция программного обеспечения / А.С. Алексеев, В.И. Гололобов, В.П. Ильин, В.И. Карначук // Новосибирск, препринт N 821; ВЦ СО АН СССР. — 1988. — 43 с.
16. Городничев, М.А. HPC Community cloud: эффективная организация работы научно-образовательных суперкомпьютерных центров / М.А. Городничев, В.Э. Малышкин, Ю.Г. Медведев // Научный вестник НГТУ. — 2013. — № 3(52). — P. 92–96.
17. Каляев, И.А. Семейство реконфигурируемых вычислительных систем с высокой производительностью / И.А. Каляев, И.И. Левин // Вычислительные методы и программирование. — 2009. — Т. 10. — № 1. — С. 207–214.

Ильин Валерий Павлович, д.ф.-м.н., г.н.с., Институт вычислительной математики и математической геофизики СО РАН, профессор, Новосибирский Государственный Университет (Новосибирск, Российская Федерация), ilin@sscc.ru

Поступила в редакцию 14 февраля 2014 г.

*Bulletin of the South Ural State University
Series “Computational Mathematics and Software Engineering”
2014, vol. 3, no. 1, pp. 68–79*

ON SOME “OVER-CLOUD” PROBLEMS OF MATHEMATICAL MODELLING

V.P. Il'in, Institute of Computational Mathematics and Mathematical Geophysics of Siberian Branch of the Russian Academy of Sciences, Novosibirsk State University (Novosibirsk, Russian Federation)

The development of integrated applied software environment for mathematical modelling, which is oriented onto solving interdiscipline direct and inverse problems by the wide circle of users in the framework of cloud computing at the supercomputers of petaflops level stands the big set of issues of methodical, technological and organizational types. In this paper, we consider the

conception of basic system of modelling (BSM), whose functionality is supposed in the framework of collective using computing center (CUCC). BSM presents the integrated software environment which supports all main technological stages of large scale computational experiments, adapted to evolution of the problem models and algorithms, as well as to architectures of multi-processor computational systems (MCS), and intended to the long life circle, which is provided by the efficient constructing of applied program packages (APPs). The achievement of the total high performance of CUCC is considered as the constrained optimization problem for the schedule of computational tasks, under taking into account the necessary resources for the particular applications.

Keywords: scalable parallelism, cloud computing, mathematical modelling, multiprocessor computational system, basic system of modelling, applied software.

References

1. DATA CENTER: URL: www.datacenterknowledge.com
2. Il'in V.P., Skopin I.N. *Technologii vychislitel'nogo programmirovaniya.*//Programmirovaniye. – 2011. – №4. – P. 53-72.
3. Golubeva L.A., Il'in V.P., Kozyrev A.N. *O programmnykh technologiakh v geometricheskikh aspektakh matematicheskogo modelirovaniya.*//Vestnik NGU, seria "Informatsionnye technologii". – 2012. – Vol. 10. – № 2. – P. 25-33.
4. Il'in V.P. *DELAUNAY: technologicheskaya sreda generatsii setok.*//SibJIM. – 2013. – Vol.16, №2(54). – P. 83-97.
5. OpenFOAM® – The Open Source Computational Fluid Dynamics (CFD) Toolbox: URL: www.openfoam.com.
6. DUNE Numerics: URL: www.dune-project.org
7. OPEN CASCADE: URL: www.opencascade.org
8. SALOME: URL: www.salome-platform.org
9. IESP: URL: www.exascale.org/iesp
10. Chesbro G. *Otkrytye innovatsii.* – M., izd. "Pokolenie". – 2007. – 333с.
11. OPEN NOVATION: URL: www.opennovation.org
12. ANSYS – Simulation Driven Product Development: URL: www.ansys.com
13. MSC Nastran – Industry Leading Multidisciplinary FEA: URL: www.mssoftware.com/product/msc-nastran
14. Kleppe A. *Software Language Engineering: Creating Domain-Specific Language Using Metamodels.*//N.Y. Addison-Wesley. – 2008. – 207 p.
15. Aleksseev A.S., Gololobov V.I., Il'in V.P., Karnachuk V.I. *Kompleksnyi centr matematicheskogo modelirovaniya: kontseptsya programmno obespecheniya.*//Novosibirsk, preprint №821; VC SOAN SSSR. – 1988. – 43pp.
16. Gorodnichev M.A., Malyshkin V.E., Medvedev Yu. G. *HPC Community Cloud: effektivnaya organizatsiya raboty nauchno-obrazovatelnykh centrov.*//Nauchnyi Vestnik NGTU. – 2013. – №3(52). – P. 92-96.
17. Kalyaev I.A., Levin I.I. *Semeistvo rekonfiguriruemyykh vychislitelnykh system s vysokoi proizvoditelnostyu.*//Vychislitelnye Metody i Programmirovaniye. – 2009. – Vol. 10. – № 1. – P.207-214.

Received 14 February 2013

МОДЕЛИРОВАНИЕ ЭКСТРЕМАЛЬНОГО НАВОДНЕНИЯ В ДЕЛЬТЕ ДОНА НА МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ¹

В.Н. Дацюк, Л.А. Крукиер, А.Л. Чикин, Л.Г. Чикина

По материалам ежедневных гидрометеорологических наблюдений на береговой базе Южного научного центра РАН в период с 20 марта по 26 марта 2013 года, проведено восстановление картины аномального затопления дельты Дона. Расчеты уровня воды и объема поступающей в дельту Дона воды проводились с помощью двухслойной математической модели. Расчеты проводились на многопроцессорных вычислительных системах, установленных в Южном федеральном университете.

Ключевые слова: математическая модель, нагон воды, многопроцессорная вычислительная система.

Введение

23–24 марта 2013 г. в результате сильного штормового нагона воды из акватории Азовского моря значительная территория в дельте Дона чрезвычайно быстро была затоплена водой, в 20 населенных пунктах более 2,4 тыс. домовладений и свыше 5,3 тыс. человек оказались пострадавшими от наводнения. Материальные потери для населения и экономики региона оцениваются суммой более 500 млн. руб. Ветровая ситуация в период с 20 марта по 26 марта 2013 года складывалась таким образом, что ветер существенно менял направление и скорость: первую половину 23 марта дул ветер восточных направлений 3–11 м/с, к 19 часам направление ветров разворачивается на юг, усиливаясь до 5–12 м/с, после полуночи 24 марта ветра приняли западное направление со скоростью до 15 м/с. К 14–17 часам 24 марта порывы ветра достигали 20–22 м/с, что привело к очень быстрому затоплению прибрежных территорий, строений и объектов в населенных пунктах. Уровень воды поднимался со скоростью 11–12 см/час. К 18 часам 24 марта наблюдался максимальный уровень воды (280 см) по данным самописца, установленном на посту в п. Донской.

Ситуацию с затоплением усугубил повышенный (до 400 м³/с) попуск воды через гидроузел Цимлянского водохранилища в период с 21 по 25 марта, при этом дошедшая до низовьев Дона за двое суток волна половодья могла быть причиной дополнительного подпора вод, поступающих из Таганрогского залива.

По метеорологическим наблюдениям, полученным с береговой научно-экспедиционной базы Южного научного центра РАН на период с 20 марта по 26 марта 2013 года было промоделировано развитие нагонного явления в дельте Дона.

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

1. Постановка задачи

При моделировании использовалась двухслойная математическая модель гидродинамики водоема, достаточно адекватно описывающая ветровые течения в Азовском море в целом, и в Таганрогском заливе в частности [3]. Данные водоемы имеют как относительно глубоководные районы, так и обширные районы мелководья.

Суть нашего подхода состоит в разбиении исходной трехмерной области расчета Ω (водной толще водоема) ограниченной сверху акваториальной, а снизу донной поверхностями на две — глубоководную и мелководную части. Для декомпозиции пространственной области моделирования Ω проведем горизонтальную секущую плоскость P , отстоящую от невозмущенной поверхности водоема P_0 на глубине, равной максимальной глубине мелководья (рис. 1). Таким образом плоскость P разделила исходную область на две подобласти: верхний слой Ω_1 (слой I) — все мелководье и верхняя часть глубоководного слоя, и глубоководный слой Ω_2 (слой II). Предполагается, что эффект осушения из-за стока воды может присутствовать только в мелководных районах.

Границы расчетной области Ω могут быть твердыми $\partial\Omega_T$ (донная поверхность, переходящая в береговую линию), участками втекания или вытекания воды $\partial\Omega_R$, свободной поверхностью $\partial\Omega_S$.

Считается, что на движение воды в слое I влияет ветер, а движение в слое II инициируется как градиентами давления, так и движением слоя I.

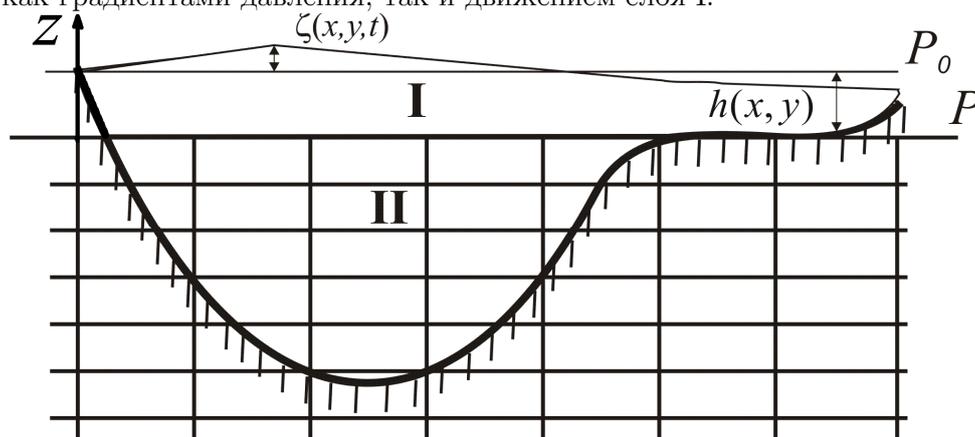


Рис. 1. Вертикальный разрез водоема с обширным мелководьем

Считаем, что слой I достаточно мелкий (значения возможных возмущений уровня воды и глубины слоя близки), а u и v не зависят от z . Движение воды в слое I описывается уравнениями мелкой воды:

$$\frac{du_s}{dt} - \Omega v_s = -g \frac{\partial \zeta}{\partial x} + v_{xy} \left(\frac{\partial^2 u_s}{\partial x^2} + \frac{\partial^2 u_s}{\partial y^2} \right) + \frac{\tau_{sx}}{H} - \frac{\tau_{bx}}{H} + F_x(x, y), \quad (1)$$

$$\frac{dv_s}{dt} + \Omega u_s = -g \frac{\partial \zeta}{\partial y} + v_{xy} \left(\frac{\partial^2 v_s}{\partial x^2} + \frac{\partial^2 v_s}{\partial y^2} \right) + \frac{\tau_{sy}}{H} - \frac{\tau_{by}}{H} + F_y(x, y), \quad (2)$$

$$\frac{\partial \zeta}{\partial t} + \frac{\partial H u_s}{\partial x} + \frac{\partial H v_s}{\partial y} = 0. \quad (3)$$

Здесь $H = h + \zeta$; $h = h(x, y)$ — глубина мелководного слоя; $u_s = u_s(x, y, t)$, $v_s = v_s(x, y, t)$ — скорости в слое I; функции $F_x(x, y)$ и $F_y(x, y)$ описывают взаимодействие верхнего и

нижнего слоев между собой; $\zeta = \zeta(x, y, t)$ – возмущение уровня воды; Ω – коэффициент Кориолиса; τ_{sx}, τ_{sy} – проекции на оси ОХ и ОУ силы трения ветра о поверхность водоема; τ_{bx}, τ_{by} – проекции на оси ОХ и ОУ силы трения жидкости о дно (или нижний слой воды). Эти величины зависят от скорости ветра $\bar{W}_B \{W_x, W_y\}$ и течения $\bar{W}_T \{u_s, v_s\}$ и определяются так:

$$\bar{\tau}_s = \gamma |\bar{W}_B| \bar{W}_B, \quad \bar{\tau}_b = \beta |\bar{W}_T| \bar{W}_T,$$

где $\beta(x, y)$ – коэффициент трения верхнего слоя жидкости о дно (или о глубоководный слой); γ – коэффициент трения ветра о слой I.

Движение воды в глубоководном слое II описывается системой, состоящей из уравнений количества движения, уравнения неразрывности среды и уравнения гидростатического давления:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} - \Omega v = -\frac{1}{\rho} \frac{\partial p}{\partial x} + v_{xy} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{\partial}{\partial z} \left(v_z \frac{\partial u}{\partial z} \right), \quad (4)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} + \Omega u = -\frac{1}{\rho} \frac{\partial p}{\partial y} + v_{xy} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + \frac{\partial}{\partial z} \left(v_z \frac{\partial v}{\partial z} \right), \quad (5)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \quad (6)$$

$$p = g\rho(\zeta - z) + p_a.$$

Здесь $u = u(x, y, z, t), v = v(x, y, z, t), w = w(x, y, z, t)$ – компоненты вектора скорости; $p(x, y, z, t)$ – давление; x, y, z, t – пространственные переменные и время соответственно; $p_a = p_a(x, y)$ – атмосферное давление; $v_{xy}, v_z(z)$ – коэффициенты горизонтальной и вертикальной вязкости соответственно; ρ – плотность воды; $g = 9,8 \text{ м/с}^2$ – ускорение силы тяжести.

Граничные условия на твердой границе $\partial\Omega_T$ задаются условиями скольжения:

$$\mathbf{V}_n|_{\partial\Omega_T} = 0, \quad \frac{\partial \mathbf{V}_\tau}{\partial n}|_{\partial\Omega_T} = 0,$$

где \mathbf{V}_n – нормальная составляющая вектора скорости, \mathbf{V}_τ – касательная составляющая вектора скорости. В местах втекания или вытекания воды $\partial\Omega_R$ задаются соответствующие значения скоростей $u|_{\partial\Omega_R} = u_1, v|_{\partial\Omega_R} = v_1, u_s|_{\partial\Omega_R} = u_{s1}, v_s|_{\partial\Omega_R} = v_{s1}$.

На границе между слоями $\partial\Omega_I$ ставится условие равенства скоростей

$$u|_{\partial\Omega_I} = u_s, v|_{\partial\Omega_I} = v_s.$$

Функции $F_x(x, y)$ и $F_y(x, y)$, описывающие взаимодействие I и II слоя, задаются следующим образом:

$$F_x(x, y) = \frac{uw}{H}, \quad F_y(x, y) = \frac{vw}{H}.$$

В качестве начальных данных можно задавать известное распределение скоростей и уровня воды

$$u|_{t=0} = u^0, \quad u_s|_{t=0} = u_s^0, \quad v|_{t=0} = v^0, \quad v_s|_{t=0} = v_s^0, \quad w|_{t=0} = w^0, \quad \zeta|_{t=0} = \zeta^0$$

или считать эти значения нулевыми.

Уравнения модели решаются конечно-разностными методами. Алгоритм вычисления параметров течения воды на $(n+1)$ -м временном слое основан на том принципе, что каждое уравнение является «определяющим» для своего неизвестного. Все остальные переменные считаются известными и берутся с n -го слоя. При конечно-разностной аппроксимации уравнений количества движения используются неявные "противопотоковые" схемы. Перепад уровня воды и вертикальная компонента скорости определяются из разностных аналогов уравнений дифференциальных уравнений.

Для решения систем линейных алгебраических уравнений, возникающих при дискретизации исходных дифференциальных уравнений, использовалась библиотека параллельных подпрограмм Aztec. В этой библиотеке реализован набор итерационных методов Крылова для решения систем линейных алгебраических уравнений с разреженными матрицами. Распараллеливание выполнено в парадигме «распределенной памяти» с использованием коммуникационной библиотеки MPI. Такой подход не только ускоряет вычисления, но значительно снижает требования к объему оперативной памяти на вычислительных узлах. При увеличении размера сетки при дискретизации решаемой системы уравнений достаточно просто увеличить количество вычислительных узлов для решения задачи. Особенностью итерационных методов является многократное использование операции умножения матрицы на вектор. Профилирование программы показало, что на эту процедуру приходится более 50% всех вычислительных затрат [1]. Поэтому оптимизации параллельной версии этой подпрограммы для минимизации объемов пересылаемых данных было уделено большое внимание. Тем не менее, из-за большого удельного веса коммуникационных операций при работе с разреженными матрицами трудно рассчитывать на высокую степень масштабируемости этой процедуры и соответственно всей программы в целом. Кроме того, прямое тестирование подтвердило, что использование этих методов предъявляет повышенные требования к коммуникационной среде.

На задаче расчета течений было протестировано время счета на трех вычислительных системах [2]:

- 1) на кластере IBMX с 2-х ядерными процессорами Intel Xeon 5160 и коммуникационной сетью DDR Infiniband;
- 2) на кластере INFINI с процессорами Intel Pentium 4 3.4 ГГц и коммуникационной сетью SDR Infiniband;
- 3) на рабочей станции QUAD с 4-х ядерным процессором Q6600 с тактовой частотой 2.4 ГГц и оперативной памятью 4Гбайт. Использовалась разностная сетка с максимальным числом неизвестных $n=1700000$. При тестировании расчет проводился для 200 временных шагов.

Как и следовало ожидать, наибольшее ускорение (Sp) прослеживается на кластере IBMX, с коммуникационной сетью DDR Infiniband (см. таблицу). Линейное ускорение на этой платформе сохраняется для числа узлов. В то время как, на кластере с одно-

кратной скоростью Infiniband ускорение начинает отклоняться от линейного уже для $n_p \leq 4$. И совершенно неоправданным представляется запуск нескольких MPI процессов на многоядерных вычислительных узлах (рабочая станция QUAD).

Сравнение производительности различных вычислительных платформ на задаче расчета ветровых течений в Керченском проливе.

Np	INFIN	Sp	IBM	Sp	QUA	Sp
	I (с)		X (с)		D (с)	
1	3896	1,0	3828	1,0	2740	1,0
2	2132	1,8	2015	1,9	2154	1,3
3	1542	2,5	1376	2,8	2865	0,95
4	1212	3,2	1066	3,6	2920	0,93
5	1029	3,8	881	4,3	-	
6	903	4,3	758	5,1	-	
7	802	4,9	681	5,6	-	

Численное исследование показало, что при использовании технологии MPI на вычислительной системе QUAD с 4-х ядерным процессором производительность немного возрастает при подключении 2-го ядра, а затем падает при подключении последующих. В то время, как на системах с распределенной памятью INFINI и IBMX наблюдается хороший рост ускорения при увеличении числа вычислительных узлов. Заметим, что на вычислительном кластере IBMX с 2-ядерным процессором на каждом узле запускался только один счетный процесс. Использование второго ядра, на малом числе узлов приводило к небольшому ускорению работы программы, а при увеличении числа узлов так же, как и на системе QUAD к замедлению работы программы.

2. Анализ результатов расчета

Качество алгоритма оценивается совпадением расчетных значений с натурными данными, полученными на метеопосту в п. Донской (рис. 2).

Расчет проводился с шагом по времени $dt = 5$ минут, и данные снимались через каждые 3 часа. Видно, что модель достаточно хорошо описывает развитие гидрологической ситуации в данном районе.

Для расчета объема поступающей в дельту Дона воды (куб.м/с) из Таганрогского залива было рассмотрено поперечное сечение восточной части залива длиной около 30 км. Считалось, что на затопление расходуется только нагонная составляющая перепада уровня, которая с учетом скорости течения давала объемный расход поступающей воды.

На рис. 3 приведен график изменения объема поступающей из Таганрогского залива воды в дельту Дона.

Видно, что максимальный объем воды поступал в ночь с 23 на 24 марта и составил около 10 тыс. куб. м в секунду. Затем поступление стало резко уменьшаться, но все равно скорость поступления воды оставалось положительной, что вело к продолжению процесса затопления. И только к вечеру 24 марта поступление воды прекратилось, и во-

да стала убывать. Вечером 25 марта расход убывающей в Таганрогский залив воды составил примерно 5 тыс. куб.м/с.

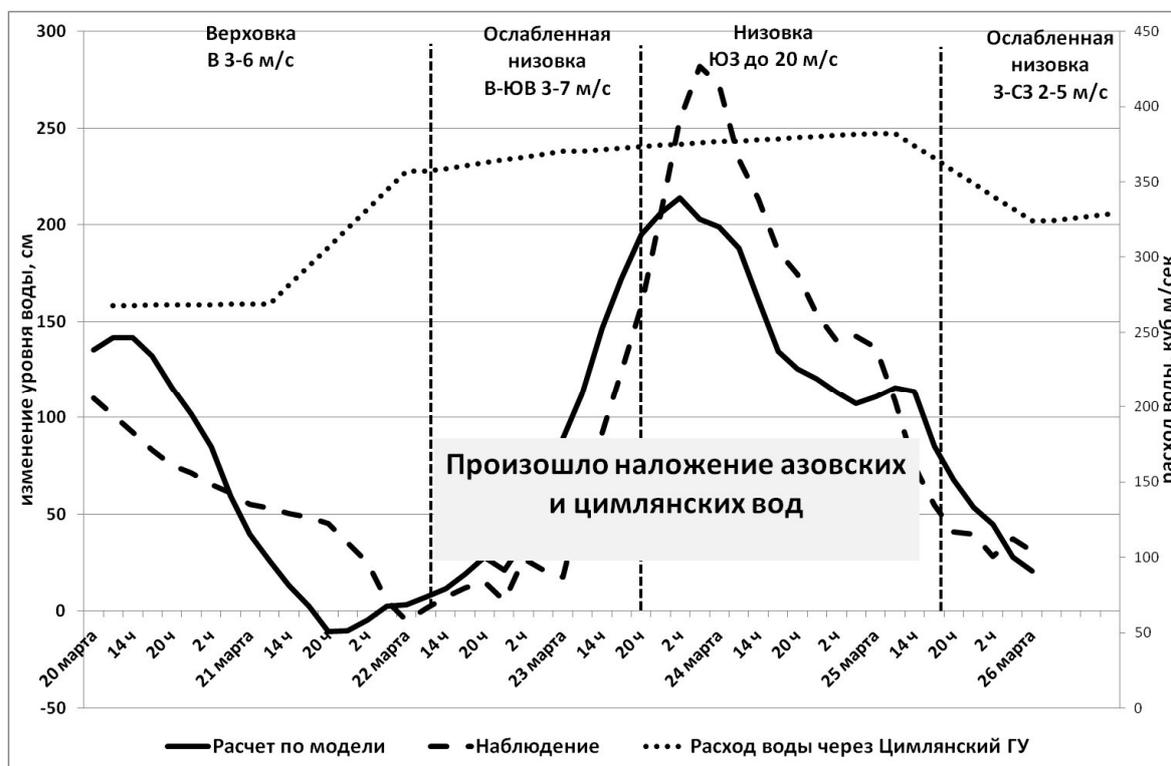


Рис. 2. Изменение относительного уровня воды на период с 20 по 26 марта 2013 года на посту «Кагальник»

С помощью моделирования установлено, что резкая смена сгонного эффекта на нагонный увеличивает скорость поступления воды, а, следовательно, и степень затопления дельты. Сравнивались две ситуации:

а) начало нагона при юго-западном ветре после штилевой погоды и

б) начало нагона после сгонного явления, то есть после действия восточного ветра.

Сгон моделировался при скорости восточного ветра 9 м/с, который затем изменялся на ветер юго-западного направления скоростью 15 м/с.

В случае предварительного сгона значительно увеличивается скорость нагонного течения, а соответственно, и объем поступающей в дельту воды. Хотя в обоих случаях затопление дельты длился 22–23 часа, что соответствует динамике повышения уровня 23–24 марта, изменения уровня воды существенно различаются. Если нагон начинается при штилевой погоде, то уровень воды поднимается от нулевой отметки, соответствующей естественной глубине водоема (рис. 4), и дальше монотонно возрастает по мере поступления излишков воды. Если перед нагоном наблюдался сгон воды, то сначала уровень отрицателен (что характерно для сгонного явления), но затем быстро увеличивается и, в конце концов, становится выше нагона после штиля. Этот результат показывает, что при резкой смене восточного ветра на западный затопление происходит быстрее и оно сильнее по масштабу, чем при постоянно действующим западном ветре.

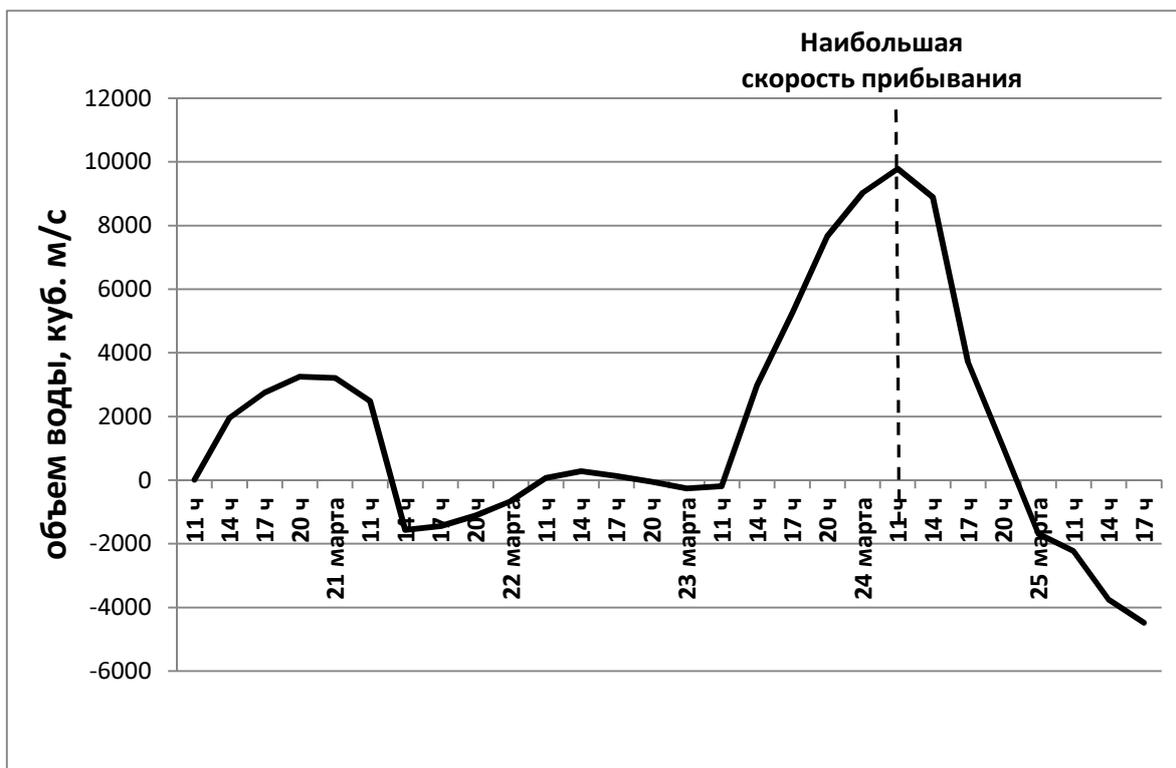


Рис. 3. Изменение объемного расхода воды на период с 20 по 26 марта 2013 года

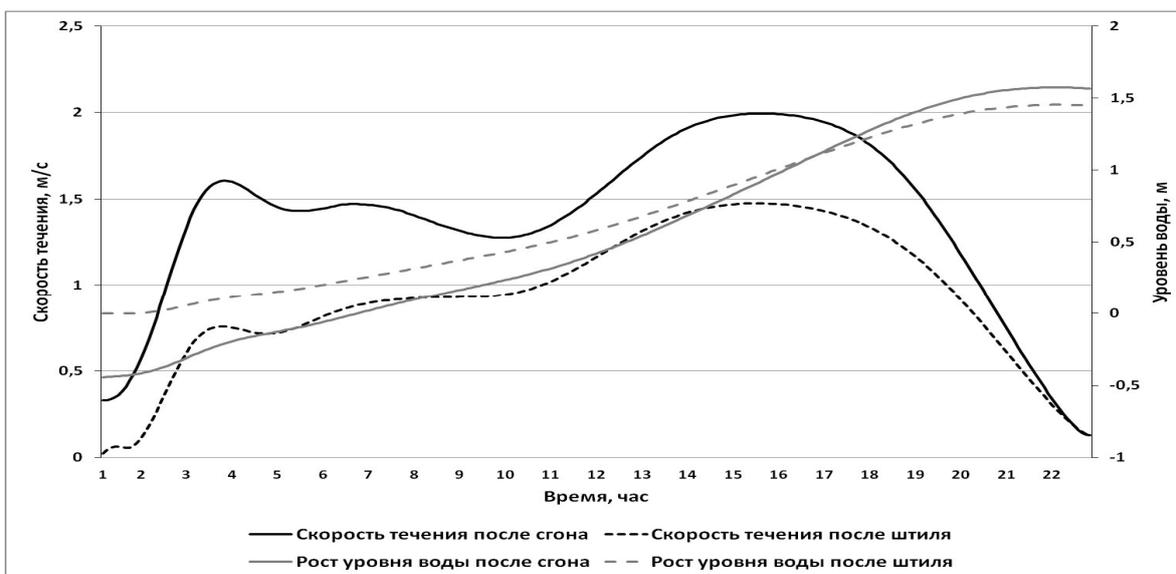


Рис. 4. Поведение скорости течения и уровня воды при различных начальных условиях нагона

Заключение

Сравнение результатов расчета с наблюдаемыми данными показало, что представленная математическая модель достаточно адекватно описывает гидродинамику течений в северо-восточной части Таганрогского залива, а также сгонно-нагонные явления. Данная модель позволяет рассчитывать ветровые течения и сгоны-нагоны в случае изменения береговой линии, что позволит в дальнейшем прогнозировать степень опасности затопления дельты Дона.

Литература

1. Дацюк В.Н., Крукиер Л.А., Чикин А.Л. Реализация на высокопроизводительных вычислительных системах математической модели ветровых течений в Керченском проливе // Вестник УГАТУ, Т.15, № 5 (45), 2011. — С.155–160.
2. Букатов А.А., Дацюк В.Н., Дацюк О.В., Крукиер Л.А. О реальной производительности вычислительных кластеров с многоядерными узлами // Известия вузов. Северо-Кавказский регион. Естественные науки. Спецвыпуск «Теоретические и прикладные проблемы математики, экономики, образования». 2011. — С. 9–15.
3. Chikin A.L. A technique for evaluating flow parameters in water bodies with a highly heterogeneous depth. Water Resources, 2005, Vol. 32, No. 1, January. — P. 50–55.

Дацюк Виктор Николаевич, к.ф.-м.н., начальник отдела, Южно-Российский региональный центр информатизации, г. Ростов-на-Дону, Россия, dvn@sfedu.ru

Крукиер Лев Абрамович, д.ф.-м.н., профессор, директор Южно-Российский региональный центр информатизации, г. Ростов-на-Дону, Россия, krukier@sfedu.ru

Чикин Алексей Львович, д.ф.-м.н., г.н.с., Институт аридных зон Южного научного центра РАН, г. Ростов-на-Дону, Россия, chikin@sfedu.ru

Чикина Любовь Григорьевна, д.ф.-м.н., профессор, Южный федеральный университет, г. Ростов-на-Дону, Россия, lchikina@sfedu.ru

Поступила в редакцию 17 февраля 2014 г.

*Bulletin of the South Ural State University
Series “Computational Mathematics and Software Engineering”
2014, vol. 3, no. 1, pp. 80–88*

MODELING OF EXTREME FLOODS IN THE DELTA OF DON RIVER ON THE MULTIPROCESSOR COMPUTER SYSTEMS

V.N. Datsuk, Southern Federal University (Rostov-on-Don, Russian Federation),

L.A. Krukier, Southern Federal University (Rostov-on-Don, Russian Federation),

A.L. Chikin, Institute of Arid Zones Southern Scientific Centre RAS (Rostov-on-Don, Russian Federation),

L.G. Chikina, Southern Federal University (Rostov-on-Don, Russian Federation),

According to the materials of daily meteorological observations in the period from March 20 to March 26, 2013 the restoration of abnormal flooding Don delta were carried out. For calculations of water level and volume of incoming water into the delta of the Don the two-layer mathematical model were carried out. The calculations were carried on multiprocessor computer systems installed in the Southern Federal University.

Keywords: Mathematical model, surge, multiprocessor computer system.

References

1. Datsuk V.N., Krukier L.A., Chikin A.L. realizatsiya na VVS matematicheskoy modeli vetrovih techeniy v Kerchenskom prolive [Implementation on High Performance Computing Systems mathematical model of the wind currents in the Kerch Strait]. Vestnik UGATU [Vestnik Ufa State Aviation Technical University]. 2011. Vol. 15, No. 5. P. 155–160
2. Bukatov A.A., Datsuk V.N., Datsuk O.V. O real'noy proizvoditelnosti vichislitelnih klasterov s mnogoyadernimi uzlamy [About the actual performance computing clusters of multicore nodes]. Izvestiya vuzov. Severo-Kavkazskiy region. Estestvennye nauky. Spetsvipusk "Teoreticheskie i prikladnie problemi matematiki, ekonomiki, obrazovaniya" [Izvestiya Vuzov. North Caucasus region. Natural sciences. Special issue of "Theoretical and applied problems of mathematics, economics, education"]. 2011. P. 9–15.
3. Chikin A.L. A technique for evaluating flow parameters in water bodies with a highly heterogeneous depth. Water Resources, 2005, Vol. 32, No. 1, January. – P. 50–55.

Received 17 February 2014

МОДЕЛИРОВАНИЕ НА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ ПРОЦЕССОВ РАСПРОСТРАНЕНИЯ МНОГОКОМПОНЕНТНЫХ ПРИМЕСЕЙ В ВОДОЕМЕ¹

И.Н. Шабас

Рассматриваются процессы переноса многокомпонентных веществ в водоеме. Примером таких веществ могут являться радионуклидные и нефтяные загрязнения природных водоемов. Полученные в результате математического моделирования численные задачи решаются на высокопроизводительных вычислительных системах (ВВС) с использованием существующих пакетов распараллеленных итерационных методов. В качестве исследуемого водоема рассматривается Азовское море.

Ключевые слова: математическое моделирование, конвективно-диффузионный перенос, вычислительный эксперимент, высокопроизводительные вычислительные системы.

Введение

В настоящее время биосфера Земли подвергается нарастающему антропогенному воздействию. При этом можно выделить несколько наиболее существенных процессов, любой из которых не улучшает экологическую ситуацию на планете.

Наиболее масштабным и значительным является химическое загрязнение окружающей среды несвойственными ей веществами химической природы. Химическое заражение почв, засорение воздуха и увеличивающееся загрязнение вод Мирового океана – все эти факторы оказывают заметное воздействие на процессы, происходящие в биосфере.

Многие химические вещества, попадающие в водоемы в результате техногенных аварий, а также в процессе функционирования промышленных предприятий, представляют собой не однородные, а многофазные, многофракционные вещества или многокомпонентные смеси. К таким веществам, например, относятся нефтяные загрязнения, тяжелые металлы и радионуклиды.

1. Постановка задачи

В области $\bar{\Omega} \times T$, $\bar{\Omega} = \Omega \cup \Gamma$ рассматривается система трехмерных уравнений, описывающая процессы переноса в многофазной среде [1]:

$$\frac{\partial \bar{S}}{\partial t} - \sum_{i=1}^3 \frac{\partial}{\partial x_i} \left(M_i \frac{\partial \bar{S}}{\partial x_i} \right) + \gamma \sum_{i=1}^3 \frac{\partial}{\partial x_i} (V_i \bar{S}) + (1 - \gamma) \sum_{i=1}^3 V_i \frac{\partial \bar{S}}{\partial x_i} + \mathcal{B}(\bar{S}) \bar{S} = \bar{f}, \quad (1)$$

$$\operatorname{div} \bar{v}_i = 0, \quad (2)$$

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

где $\bar{S} = \{S_1, S_2, \dots, S_k\}$ – вектор концентраций i -х веществ, M_1, M_2, M_3 – диагональные матрицы размера $k \times k$ с элементами $\{\mu_i^\alpha\}_{i=1}^k$, – коэффициентами турбулентной диффузии i -го вещества, $\alpha = 1, 2, 3$, V_1, V_2, V_3 – диагональные матрицы размера $k \times k$, элементами которых являются скорости движения каждой компоненты примеси по направлениям x, y и z соответственно, $\bar{v}_i = \{u_i, v_i, w_i\}$ – вектор скорости движения i -го вещества. В общем случае нелинейный оператор вида $\mathcal{B}(\bar{S})$ – описывает взаимодействие веществ в среде. Система (1) замыкается начальными

$$\bar{S}|_{t=0} = \bar{S}^0 \quad (3)$$

и смешанными краевыми условиями на границе $\Gamma = \bigcup_{i=0}^4 \Gamma_i$ области Ω [2]:

$$G \frac{\partial \bar{S}(\mathbf{x})}{\partial \bar{n}} + P \bar{S}(\mathbf{x}) = R(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \quad t > 0, \quad (4)$$

где Γ_0 – боковая непроницаемая граница, Γ_1 – участки впадения рек, Γ_2 – открытые участки границы, на которых происходит водообмен, Γ_3 – поверхность водоема, Γ_4 – дно водоема. Здесь G, P, R – диагональные матрицы размера $k \times k$, $\frac{\partial \bar{S}}{\partial \bar{n}}$ – это производная по нормали \bar{n} к границе Γ .

1.1. Радионуклиды

Примером многофазной жидкости может служить водоем, в который попало некоторое количество радионуклидов. Фазами будем называть различные состояния рассматриваемого радионуклида. Часть радионуклидов будет находиться в *растворенной фазе* C , часть вступать во взаимодействие с частицами взвеси и составлять *взвешенную фазу* C^s . Третий компонент смеси – осевшие на дно радионуклиды – образуют *донную фазу* C^b . Наличие в водоеме взвесей S обеспечивает переход растворенной фазы во взвешенную. Таким образом, вектор концентраций в системе (1) примет вид $\bar{S} = \{S, C, C^s, C^b\}$.

При описании процесса распространения радионуклидов в Азовском море была взята за основу модель, предложенная киевскими учеными под руководством М.И. Железняка [3].

1.2. Нефть

Нефтяные загрязнения водоемов – неизбежная опасность, возникающая в процессе транспортировки нефти водным путем от мест ее добычи к местам переработки.

Распространение нефти в водоеме представляет собой достаточно сложный процесс, при моделировании которого необходимо учитывать большое количество разнообразных факторов. На поведение нефти, попавшей в водоем, оказывает влияние как собственные физико-химические свойства нефти (температура кипения фракций, их плотность, вязкость), так и внешние условия окружающей среды (ветровая ситуация, температура воздуха и температура вод водоема, наличие в водоеме нефтеокисляющих бактерий, солености водоема, солнечной радиации и т.п.).

В процессе распространения нефти вначале преобладают процессы растекания нефтяного пятна [4, 5], параллельно с этими процессами начинает происходить неизбежная деструкция нефти, под воздействием внешних природных факторов. К процессу деструкции нефти можно отнести испарение легких нефтяных фракций, их растворение в водах водоема, а также эмульсификация и биodeградация нефти. Эти процессы оказывают влияние

на исходную плотность и вязкость нефти – они увеличиваются, а суммарное поверхностное натяжение на границе раздела вода – нефть – воздух убывает. В дальнейшем, остающиеся тяжелые фракции нефти могут оказаться плотнее воды и осесть на дно водоема.

На определенном этапе поверхностное натяжение меняет знак и растекание прекращается.

Поведение пятна нефти на поверхности водоема описывается уравнением конвекции-диффузии [6–8]:

$$\begin{aligned} \frac{\partial h}{\partial t} + \bar{\nabla}(\bar{v}h) - \bar{\nabla}(D\bar{\nabla}vh) &= R_h, \\ \bar{v} &= \left(u_x + \frac{\tau_x^w}{C_f}, u_y + \frac{\tau_y^w}{C_f} \right), \\ D &= \frac{gh^2(\rho_w - \rho_{oil})}{\rho_w C_f}, \end{aligned}$$

где h – толщина нефти, \bar{v} – скорость дрейфа пленки, $\frac{\tau_x^w}{C_f}$ – напряжение сдвига из-за ветра, D – функция диффузии распространения пятна нефти, C_f – коэффициент трения между нефтяной пленкой и поверхностью воды ($0.02 \text{ кг/м}^2 \text{ с}$), R_h – источниковая (стоковая) функция, g – ускорение свободного падения, $\bar{\nabla} = (\partial/\partial x, \partial/\partial y)$.

Полученные системы уравнений решаются конечно-разностными методами с использованием неявных схем. В расчетной области строится прямоугольная равномерная по всем направлениям сетка. Шаг сетки по горизонтали составляет 2500 м, а по вертикали – 1 м. Предполагается, что поле скоростей известно на каждом временном шаге.

В результате конечно-разностной аппроксимации получаются системы линейных уравнений с семидиагональной матрицей в случае радионуклидов и с пятидиагональной матрицей в случае модели нефтяных загрязнений.

2. Параллельные вычисления

Решение задачи проводится на многопроцессорных вычислительных системах с распределенной памятью в среде параллельного программирования MPI. В суперкомпьютерном центре ЮФУ существуют 4 вычислительных кластера, которые работают под управлением общего управляющего сервера:

- **INFINI** – Linux-кластер, состоящий из 20 вычислительных узлов, соединенных скоростной коммуникационной сетью SDR Infiniband. Каждый вычислительный узел представляет собой компьютер с процессором Intel Pentium 4 3.4 ГГц и оперативной памятью DDR2 2 Гб.
- **IBMX** – Linux-кластер, состоящий из 13 вычислительных узлов, соединенных скоростной коммуникационной сетью DDR Infiniband. Каждый вычислительный узел представляет собой компьютер с одним 2-х ядерным процессором Intel Xeon 5160 с тактовой частотой 3.0 ГГц и оперативной памятью 8 Гб.
- **LINUX** – вычислительный кластер из 10 узлов, соединенных вычислительной сетью Gigabit Ethernet. Каждый из узлов представляет собой компьютер с процессором Pentium 4 2.4 ГГц, с 512 Мб оперативной памяти.
- **WSD** – кластер из 8-ми рабочих станций DELL с двух ядерными процессорами Intel Core 2 Duo, оперативной памятью 4 Гб и коммуникационной сетью Gigabit Ethernet.

2.1. Библиотека параллельных методов Aztec

Для реализации возможностей, которые предоставляют многопроцессорные системы, необходимо использование параллельных подпрограмм, предназначенных для решения стандартных задач численных методов. На всех многопроцессорных системах суперкомпьютерного центра ЮФУ установлена одна из таких библиотек параллельных подпрограмм – Aztec, предназначенная для решения систем линейных алгебраических уравнений с разреженной матрицей. Программы, использующие этот пакет, могут выполняться на любой вычислительной системе без каких либо модификаций.

Aztec включает в себя процедуры, реализующие ряд итерационных методов Крылова – метод сопряженных градиентов (CG), обобщенный метод минимальных невязок (GMRES), квадратичный метод сопряженных градиентов (CGS), метод квазиминимальных невязок (TFQMR), метод бисопряженных градиентов (BiCGSTAB) со стабилизацией. Все методы используются совместно с различными переобуславливателями (полиномиальный метод и метод декомпозиции областей, использующий как прямой метод LU, так и неполное LU разложение в подобластях).

2.1.1. Эффективность распараллеливания

Для оценки эффективности распараллеливания было протестировано время счета метода BiCGStab на кластере INFINI на 100 временных шагах с применением различного количества ядер вычислительной системы. Результаты этих расчетов приведены в табл. 1 и на рис. 1.

Таблица 1

Сравнение времени счета метода BiCGStab на кластере INFINI на 100 временных шагах с применением различного количества узлов

Кол-во ядер	1	2	3	4	5	6	7	8	9	10	11	12
Время счета (с)	360	203	138	104	84	74	63	59	54	50	48	44
Ускорение	1,00	1,77	2,61	3,46	4,29	4,86	5,71	6,10	6,67	7,20	7,50	8,18

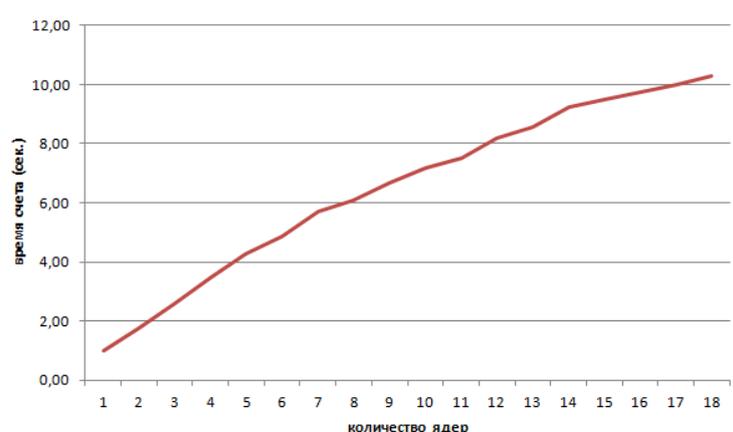


Рис. 1. Ускорение времени вычисления расчетов с применением различного количества узлов

2.1.2. Выбор эффективного итерационного метода

Для выявления наиболее эффективного для данной задачи метода из пакета Aztec был проведен ряд расчетов на одном процессоре вычислительного кластера IBMX. Для всех методов расчеты проводились для 100 временных шагов при прочих одинаковых условиях. Результаты этих тестов приведены в табл. 2.

Таблица 2

Сравнение времени счета
различных методов из пакета
Aztec на IBMX-кластере
(3,0 ГГц)

Метод	Время счета (с)
CG	328,46
GMRES	330,55
CGS	330,88
TFQMR	333,95
BiCGStab	330,73

Видно, что существенного различия во времени счета нет, однако меньшее время было затрачено при применении метода CG. Можно предполагать, что для более длительного счета, этот метод будет наиболее предпочтительным.

2.1.3. Выбор вычислительной платформы

Для тех же 100 временных шагов было протестировано время счета метода CG на различных вычислительных платформах. Результаты этих расчетов приведены в табл. 3.

Таблица 3

Сравнение времени счета метода
CG на одном процессоре
различных платформ

Платформа	Время счета (с)
IBMX	180,61
LINUX	456,14
INFINI	327,11
WSD	169,05

Таким образом, в расчетах имеет смысл применять метод CG из пакета Aztec и расчеты на кластере WSD или IBMX обеспечивают лучшее время счета.

3. Вычислительный эксперимент

Модель распространения нефтяного пятна проверялась на тестовой задаче с изменением координат залповых источников выбросов нефти, с изменением количества нефти,

попавшей в водоем в результате залпового выброса. Расчеты проводились при различных ветровых ситуациях. Результаты расчетов приведены на рисунке 2.

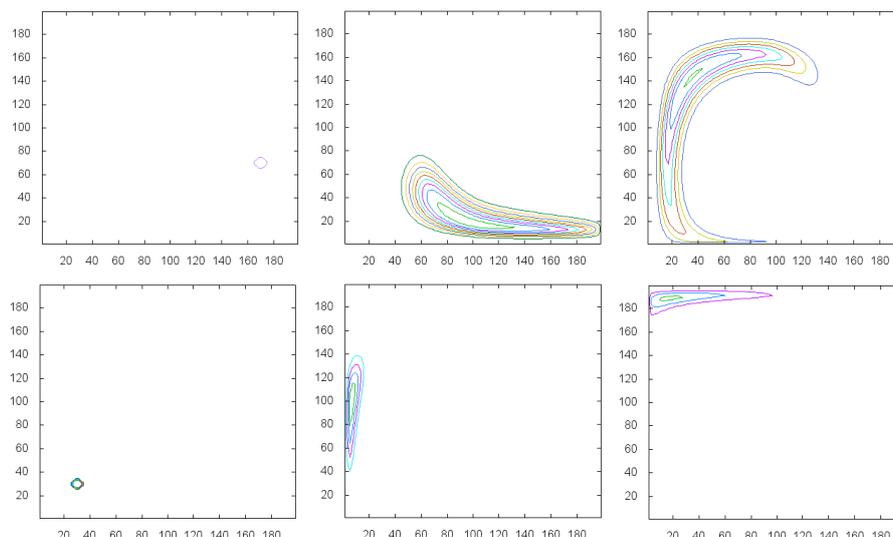


Рис. 2. Результаты моделирования дрейфа нефтяного пятна после растекания с различными координатами залпового выброса и различным начальным объемом нефти

На кластере IBMX были проведены расчеты, моделирующие аварийный выброс радионуклида в растворенной форме в районе города Ейск. Расчеты проводились с целью изучения процесса сорбции радионуклида частицами взвеси, находящейся в водах Азовского моря. В численном эксперименте течение воды формировалось под действием ветра северо-восточного направления силой 5 м/с. Сравнивались результаты расчетов, где начальное количество взвеси в одном расчете отличалось от другого в 3 раза (рис. 3). На начало расчета предполагалось отсутствие в море всех фаз радионуклидов. Исследовалась скорость образования взвешенной фазы радионуклида в зависимости от массы сорбирующего вещества.

Выводы

Результаты тестовых расчетов показали, что предложенная модель и ее программная реализация позволяют проводить расчеты, результаты которых достаточно хорошо согласуются с наблюдаемыми в природе процессами. Применение ВВС позволит в дальнейшем уточнять и усложнять предложенную модель для более реалистичного отражения ею реальных процессов.

Литература

1. Крукиер, Л.А. Неявные разностные схемы и итерационный метод их решения для одного класса систем квазилинейных уравнений / Л.А. Крукиер // Изв. Вузов. Матем. — 1979. — № 7. — С. 41–52.
2. Шабас, И.Н. Численное решение трехмерной задачи оседания вещества в Азовском море / И.Н. Шабас // Сборник трудов IX Всероссийской школы-семинара «Современные проблемы математического моделирования». — Ростов-на-Дону, Издательство РГУ, 2001. — С. 414–417.

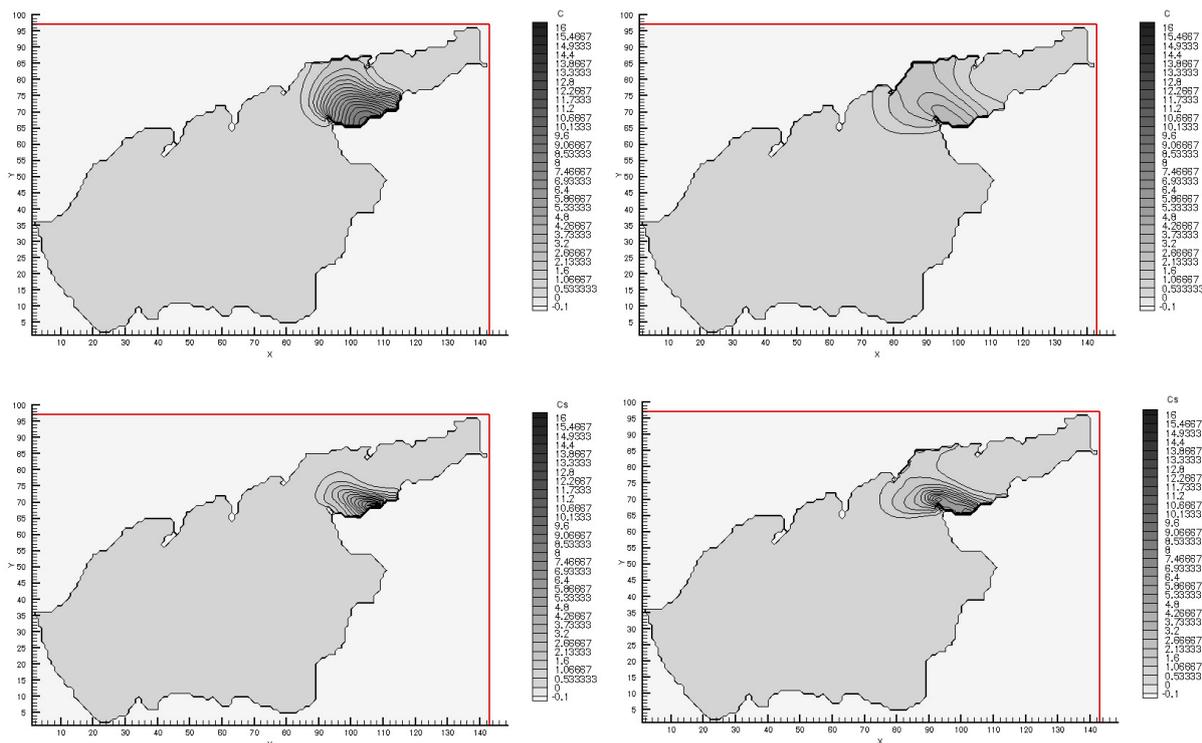


Рис. 3. Распространение растворенной (C) и взвешенной (C^s) фазы радионуклида через 48 часов (слева) и 108 часов (справа) после аварийного выброса на поверхности водоема

3. Zheleznyak, M.J. The mathematical modelling of radionuclide transport by surface water flow from the vicinity of the Chernobyl Nuclear Power Plant / M.J. Zheleznyak. // Condensed Matter Physics. — No 12. — 1997. — P. 37–50.
4. Fay, J.A. The spread of oil slicks on a calm sea. / J.A. Fay // Oil on the sea. — New-York, Plenum Press, 1969. — P. 53–63.
5. Fay, J.A. Physical processes in the spread of oil on a water surface. / J.A. Fay // Proceedings of Joint Conference on Prevention and Control of Oil Spills. — Washington, 1971. — Vol. 1971. No. 1. — P. 463–467.
6. Zadeh, E.S. Eulerian Oil Spills Model Using Finite-Volume Method with Moving Boundary and Wet-Dry Fronts / E.S. Zadeh, K. Hejazi // Modelling and Simulation in Engineering. — Vol. 2012. — Article ID 398387, 7 p.
7. Tkalich, P. A CFD solution of oil spill problems / P. Tkalich. // Environmental Modelling & Software. — Vol. 21. — 2006. — P. 271–282.
8. Three-dimensional numerical simulation for transport of oil spills in seas. / S.D. Wang, et al. // Ocean Engineering. — 2008. — Vol. 35, No. 5–6. — P. 503–510

Шабас Ирина Николаевна, кандидат физико-математических наук, старший научный сотрудник, Южно-Российский региональный центр информатизации, Южный федеральный университет (Ростов-на-Дону, Россия), shabas@sfedu.ru

Поступила в редакцию 18 февраля 2014 г.

SIMULATION OF MULTICOMPONENT POLLUTION DISTRIBUTION IN RESERVOIR ON HIGH PERFORMANCE COMPUTING SYSTEMS

I.N. Shabas, Computer Center of Southern Federal University (Rostov-on-Don, Russia)

Transport processes of multicomponent substances in the pond are considered. Examples of the substances are the radionuclide and oil pollution in natural waters. The resulting numerical problems are solved by high-performance computing systems using existing packages parallelized iterative methods. Azov Sea is considered as a test problem.

Ключевые слова: mathematical modeling, convective-diffusive transport, computational experiment, high-performance computing.

References

1. Krukier L.A. Implicit difference schemes and the iterative method for solving the class of systems of quasilinear equations // Russian Mathematics (Iz. VUZ), 1979, no 7, pp. 41–52.
2. Shabas I.N. Numerical solution of three-dimensional problem of substance concretion in the Azov Sea. Proceedings IX All-Russian Workshop "Modern Problems of Mathematical Modeling Rostov-on-Don, Publisher RSU 2001. pp. 414–417.
3. Zheleznyak M.J. The mathematical modelling of radionuclide transport by surface water flow from the vicinity of the Chernobyl Nuclear Power Plant. Condensed Matter Physics, no 12, 1997, pp. 37–50.
4. Fay J.A. The spread of oil slicks on a calm sea. In: Oil on the sea, Plenum Press. New-York, 1969, pp. 53–63.
5. Fay J.A. Physical processes in the spread of oil on a water surface. In: Proc. of h- o Joint Conf. on prevention and control of oil spills. Washington, 1971 (cit. N8).
6. Zadeh E.S., Hejazi K. Eulerian Oil Spills Model Using Finite-Volume Method with Moving Boundary and Wet-Dry Fronts. Modelling and Simulation in Engineering Volume 2012 (2012), Article ID 398387, 7 pages.
7. Tkalich P. A CFD solution of oil spill problems. Environmental Modelling & Software, vol. 21, 2006. pp. 271–282.
8. Wang S.D., Shen Y.M., Guo Y.K., Tang J. Three-dimensional numerical simulation for transport of oil spills in seas. Ocean Engineering, vol. 35, no. 5–6, pp. 503–510, 2008.

Received 18 February 2014

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧ ДИНАМИКИ ЗАРЯЖЕННЫХ ЧАСТИЦ С УЧЕТОМ БАЛАНСИРОВКИ ВЫЧИСЛИТЕЛЬНОЙ НАГРУЗКИ¹

Е.А. Берендеев, М.А. Боронина, В.Д. Корнеев

Рассмотрены задачи динамики встречных пучков заряженных частиц в ускорителях и динамики плазменных электронов в ловушке с инверсными магнитными пробками и мультипольными магнитными стенками. Модели построены на основе метода частиц в ячейках. Такие задачи требуют большого объема вычислений и могут быть решены только с применением мощных суперЭВМ. Для равномерной и полной загрузки вычислительных узлов выполнена модификация эйлерово-лагранжевой декомпозиции в случае существенно неравномерного распределения частиц по пространству и по времени.

Ключевые слова: метод частиц в ячейках, параллельные алгоритмы, балансировка нагрузки, физика плазмы, встречные пучки, ускорители частиц.

Введение

Данная работа посвящена созданию и исследованию параллельных алгоритмов для моделирования некоторых задач физики плазмы методом частиц.

В частности, для моделирования динамики встречных пучков заряженных частиц в современных ускорителях с учетом высоких значений релятивистского фактора частиц и трехмерности задачи. При этом требуется проводить расчеты на сетках порядка $512 \times 512 \times 512$ с количеством частиц в пучке $\sim 10^{10}$. В основе существующих стандартных численных моделей и алгоритмов, направленных на решение задач с высокими релятивистскими факторами (10^3 и более), лежит разделение пучка вдоль оси коллективного движения на слои частиц. При этом взаимодействие встречных пучков сводится к попарному взаимодействию слоев: частицы одного слоя через поле поперечных сил влияют на динамику частиц другого. К настоящему времени созданы и параллельные алгоритмы решения задач динамики заряженных частиц в ускорителях. Обычно каждый процессор отвечает за свой слой частиц или за несколько слоев, в некоторых случаях имеется динамическая балансировка загрузки процессоров. Но используемый квазитрехмерный подход даже с учетом распараллеливания затрудняет учет продольных эффектов при критически высокой плотности пучка, когда за короткий промежуток времени пучок деформируется или даже разрушается, а также возникают трудности моделирования таким способом при сравнительно большом угле пересечения пучков (~ 20 мрад). Существенно нелинейное распределение плотности пучка (по гауссу с условием фокусировки), значительное изменение формы пучка при пролете через область (форма песочных часов), требования к количеству частиц пучка при заданной трехмерной сетке [1] приводят к необходимости создания кода с балансировкой нагрузки процессоров.

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

Аналогичные требования к размеру сетки и количеству частиц предъявляет задача динамики плазменных электронов в ловушке с инверсными магнитными пробками и мультипольными магнитными стенками. Величина магнитного поля внутри и на торцах ловушки существенно отличается. При этом, чтобы точно воспроизвести движение частиц в сильном магнитном поле, необходимо использовать достаточно мелкий временной шаг. В слабом магнитном поле величину временного шага можно увеличить и тем самым ускорить вычисления, т.к. траектории частиц в этом случае достаточно гладкие. Однако при построении параллельного алгоритма необходимо учитывать, что за реальный промежуток времени разные частицы совершают движение за разное количество временных шагов. Таким образом, объем вычислений для каждой частицы будет разный, даже если частиц на каждом процессоре поровну.

Рассмотренные задачи обладают существенной неоднородностью распределения частиц как по пространству, так и по времени. Поэтому нашей целью явилось создание параллельного алгоритма с балансировкой нагрузки на процессоры, обладающего хорошей масштабируемостью для возможности проведения численных экспериментов с использованием высокого количества частиц. В работе описана постановка задачи, используемые модели и методы, а также описание специфики распараллеливания и сравнение с другими работами мирового уровня. Представлена блок-схема алгоритма и результаты исследований его для обеих задач: время счета, эффективность распараллеливания, ускорение. Приведены примеры использования алгоритма для проведения математического моделирования соответствующих физических явлений.

1. Постановка задачи и методы решения

В задаче рассматривается движение частиц, которое происходит в вакууме в самосогласованных электромагнитных полях с учетом внешней составляющей электромагнитного поля. При этом модельный пучок представляет собой набор достаточно большого количества частиц с соответствующими параметрами [2, 3]. Такое движение моделируется с помощью кинетического уравнения Власова [4] для функции распределения частиц с положительным $f_+ = f_+(\vec{r}, \vec{p}, t)$ и отрицательным зарядом $f_- = f_-(\vec{r}, \vec{p}, t)$:

$$\frac{\partial f_{+,-}}{\partial t} + \vec{v}_{+,-} \cdot \frac{\partial f_{+,-}}{\partial \vec{r}} + \vec{F}_{+,-} \cdot \frac{\partial f_{+,-}}{\partial \vec{p}} = 0. \quad (1)$$

Сила Лоренца $\vec{F}_{+,-}$, действующая на заряженную частицу, определяется из соотношения

$$\vec{F}_{+,-} = q_{+,-} \left(\vec{E} + \frac{1}{c} [\vec{v}_{+,-}, \vec{H}] \right). \quad (2)$$

Импульс частицы $\vec{p}_{e^{+,-}}$ связан со скоростью релятивистским фактором $\gamma_{+,-}$ равенством $\vec{p}_{+,-} = \gamma_{+,-} m \vec{v}_{+,-}$, при этом $\gamma_{+,-} = 1/\sqrt{1 - v_{+,-}^2/c^2}$, где c – скорость света.

Система уравнений Максвелла, связывает между собой плотности заряда n_+ , n_- , ток \vec{j} и напряженности электрического и магнитного полей \vec{E} и \vec{H} :

$$\begin{aligned} \operatorname{rot} \vec{E} &= -\frac{1}{c} \frac{\partial \vec{H}}{\partial t}, \\ \operatorname{rot} \vec{H} &= \frac{4\pi}{c} \vec{j} + \frac{1}{c} \frac{\partial \vec{E}}{\partial t}, \\ \operatorname{div} \vec{E} &= 4\pi (n_- e^- + n_+ e^+), \\ \operatorname{div} \vec{H} &= 0. \end{aligned} \quad (3)$$

Входящие в эти уравнения плотность заряда и плотность тока определяются через моменты функции распределения частиц:

$$\begin{aligned} n_+ &= \int_V f_+ dV, \\ n_- &= \int_V f_- dV, \\ \vec{j} &= \int_V (f_+ \vec{v}_+ e^+ + f_- \vec{v}_- e^-) dV. \end{aligned} \quad (4)$$

Уравнения характеристик уравнения Власова совпадают с уравнениями движения частиц:

$$\begin{aligned} \frac{d\vec{p}_{+,-}}{dt} &= \vec{F}_{+,-}, \\ \frac{d\vec{r}_{+,-}}{dt} &= \vec{v}_{+,-}. \end{aligned} \quad (5)$$

Система уравнений Власова-Максвелла решается методом частиц в ячейках, с использованием ядра РС. Так как в уравнения входят первые производные и по времени, и по пространству, то применяется схема с перешагиванием [5].

Компоненты магнитного поля \vec{H} вычисляются в центрах ребер ячеек, образованных пространственной сеткой, а значения электрического поля \vec{E} вычисляются в серединах граней этих ячеек (рис. 1).

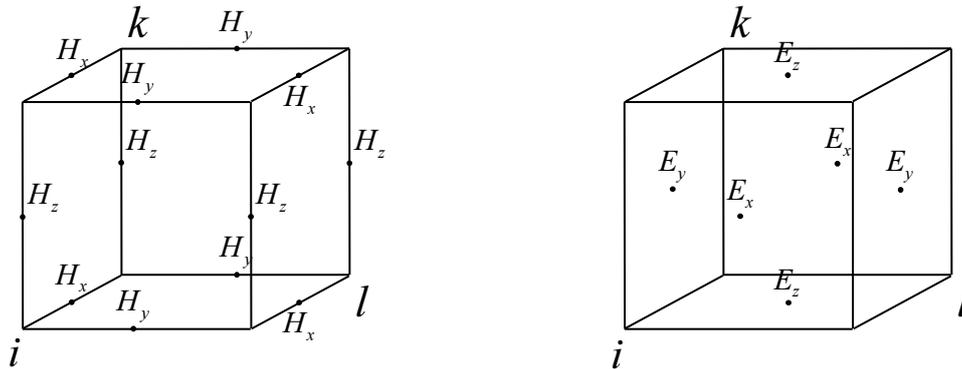


Рис. 1. Выбор узлов сетки для аппроксимации

При этом электрическое поле вычисляется в целые моменты времени $n\tau$, а магнитное поле – в дробные моменты времени $n(\tau + 1/2)$. Токи вычисляются в тех же точках пространства, что и электрическое поле, но по времени берутся в дробных узлах. Значения импульсов частиц также вычисляются в моменты времени $n(\tau + 1/2)$, а координаты – в целые. При этом все производные, участвующие в уравнениях, как по времени, так и по пространству, записываются через центральные разности, что обеспечивает при использовании этой схемы второй порядок по времени и по пространству.

Сила Лоренца определяется линейной интерполяцией по каждому направлению сил электромагнитных полей, вычисленных в каждом из ближайших к частице восьми узлов.

Уравнения Максвелла решаются по следующим конечно-разностным схемам:

$$\begin{aligned} \frac{\vec{H}^{m+\frac{1}{2}} - \vec{H}^{m-\frac{1}{2}}}{\tau} &= -rot_h \vec{E}^m \\ \frac{\vec{E}^{m+1} - \vec{E}^m}{\tau} &= \vec{j}^{m+\frac{1}{2}} + rot_h \vec{H}^{m+\frac{1}{2}}, \end{aligned} \quad (6)$$

где

$$rot_h \vec{H} = \begin{bmatrix} \frac{Hz_{i,k,l-\frac{1}{2}} - Hz_{i,k-1,l-\frac{1}{2}}}{h_y} - \frac{Hy_{i,k-\frac{1}{2},l} - Hy_{i,k-\frac{1}{2},l-1}}{h_z} \\ \frac{Hx_{i-\frac{1}{2},k,l} - Hx_{i-\frac{1}{2},k,l-1}}{h_z} - \frac{Hz_{i,k,l-\frac{1}{2}} - Hz_{i-1,k,l-\frac{1}{2}}}{h_x} \\ \frac{Hy_{i,k-\frac{1}{2},l} - Hy_{i-1,k-\frac{1}{2},l}}{h_x} - \frac{Hx_{i-\frac{1}{2},k,l} - Hx_{i-\frac{1}{2},k-1,l}}{h_y} \end{bmatrix}$$

$$rot_h \vec{E} = \begin{bmatrix} \frac{Ez_{i-\frac{1}{2},k+\frac{1}{2},l} - Ez_{i-\frac{1}{2},k-\frac{1}{2},l}}{h_y} & \frac{Ey_{i-\frac{1}{2},k,l+\frac{1}{2}} - Ey_{i-\frac{1}{2},k,l-\frac{1}{2}}}{h_z} \\ \frac{Ex_{i,k-\frac{1}{2},l+\frac{1}{2}} - Ex_{i,k-\frac{1}{2},l-\frac{1}{2}}}{h_z} & \frac{Ez_{i+\frac{1}{2},k-\frac{1}{2},l} - Ez_{i-\frac{1}{2},k-\frac{1}{2},l}}{h_x} \\ \frac{Ey_{i+\frac{1}{2},k,l-\frac{1}{2}} - Ey_{i-\frac{1}{2},k,l-\frac{1}{2}}}{h_x} & \frac{Ex_{i,k+\frac{1}{2},l-\frac{1}{2}} - Ex_{i,k-\frac{1}{2},l-\frac{1}{2}}}{h_y} \end{bmatrix} \quad (7)$$

Импульсы каждой частицы находятся с помощью выражений:

$$\frac{\vec{p}^{m+\frac{1}{2}} - \vec{p}^{m-\frac{1}{2}}}{\tau} = q \left(\vec{E}^m + \left[\frac{\vec{v}^{m+\frac{1}{2}} + \vec{v}^{m-\frac{1}{2}}}{2}, \vec{H}^m \right] \right). \quad (8)$$

Токи вычисляются по координатам частиц с использованием ядра PIC–метода по схеме, предложенной Вилласенором и Бунеманом [6]. Такой метод вычисления токов позволяет автоматически удовлетворить разностному уравнению неразрывности и, следовательно, точно выполнить разностный закон Гаусса. Это значительно уменьшает ошибки аппроксимации и делает алгоритм более устойчивым.

2. Параллельная реализация алгоритма

Метод частиц, созданный еще в 60-х годах 20 века, в настоящее время является развитым и широко применяемым методом для решения задач бесстолкновительной динамики частиц, ежегодные публикации демонстрируют постоянный интерес к этому методу.

В качестве приложения описанного выше алгоритма используются две задачи. Одна из них — получение мощных нейтральных пучков для установок управляемого термоядерного синтеза. Наиболее эффективным методом получения таких пучков является нейтрализация пучков отрицательных ионов в плазменной ловушке — мишени. В ИЯФ СО РАН предложена линейная ловушка с обратным магнитным полем [7]. Для ограничения радиальных потерь плазмы используются мультипольные магнитные стенки кольцевой геометрии. В осесимметричной ловушке с кольцевым магнитным полем отсутствует азимутальный компонент поля, а так же отсутствует стационарное азимутальное электрическое поле. Оценка и минимизация потерь плазмы в широко апертурные проходные отверстия в торцах, в которых находятся инверсные магнитные пробки, а также через цилиндрические мультипольные магнитные стенки ловушки на ее вакуумную камеру, может быть исследована только с помощью математического моделирования. При этом, наиболее полно динамика плазменных электронов может быть описана уравнением Больцмана

$$\frac{\partial f_{+,-}}{\partial t} + \vec{v}_{+,-} \frac{\partial f_{+,-}}{\partial \vec{r}} + \vec{F}_{+,-} \frac{\partial f_{+,-}}{\partial \vec{p}} = St\{f_{+,-}\}. \quad (9)$$

Здесь $St\{f_{+,-}\}$ — функция, описывающая следующие физические процессы:

- ионизация атома водорода,
- ионизация и диссоциация молекулы H_2
- диссоциативное возбуждение и диссоциативная рекомбинация H^{2+}
- диссоциативная рекомбинация D^{2+}
- перезарядка протонов на атомах водорода.

Решение уравнения Больцмана (9) можно свести к решению уравнения Власова (1) и корректировке траекторий частиц с учетом рассеяния, используя методы Монте-Карло $\frac{Df_\alpha}{Dt} = St\{f_\alpha\}$ [8].

В настоящей работе рассеяние не учитывается, оно будет являться предметом дальнейшего исследования. Таким образом, решается система, описанная в параграфе 1, но в цилиндрической системе координат. Задача рассматривается в двумерной R-Z геометрии. Особенностью данной задачи является движение частиц под воздействием магнитных полей с большими градиентами. Для ускорения расчета траекторий частиц использовался динамический шаг по времени, обеспечивающий изменение напряженности магнитного поля не более 20% за один шаг.

Другой вариант использования алгоритма — трехмерное численное моделирование эффектов встречи пучков частиц с критическими параметрами (в частности, с высокими значениями релятивистских факторов и с предельно высокой плотностью зарядов) в ускорителях. В этом случае пучок может не только сильно сжиматься, но и разрушаться, поэтому требуется исследование устойчивости пучков. С точки зрения математического моделирования в задачах с большими значениями релятивистского фактора ($\gamma \sim 10^3$) имеется существенное отличие от задач с малыми значениями ($\gamma \sim 5$). Известно, что поле движущейся заряженной частицы в лабораторной системе координат вытягивается в γ раз поперек оси движения и сокращается в γ^2 раз вдоль этой оси. Так, например, при значениях релятивистского фактора $\gamma \sim 10^3$ отношение поперечных размеров к продольному, на которых поля близки по абсолютной величине, составляет $\sim 10^9$, и использование традиционных путей решения становится невозможным.

В основе существующих стандартных численных моделей и алгоритмов, направленных на решение задач с высокими релятивистскими факторами (10^3 и более), лежит разделение пучка вдоль оси коллективного движения на слои частиц. При этом взаимодействие встречных пучков сводится к попарному взаимодействию слоев: частицы одного слоя через поле поперечных сил влияют на динамику частиц другого (коды Guinea-Pig, ODYSSEUS) [9, 10]. В большинстве существующих программных кодов, основанных на данном подходе, для получения поперечного поля используются либо формулы Бассетти – Ерскине [11], либо двумерное уравнение Пуассона с использованием быстрого преобразования Фурье (при этом граничные условия вычисляются через двумерную функцию Грина). К настоящему времени созданы и параллельные алгоритмы решения задач динамики заряженных частиц в ускорителях (коды COMBI, IMPACT, Beam Beam 3D) [12–13]. Обычно каждый процессор отвечает за свой слой частиц или за несколько слоев, в некоторых случаях имеется динамическая балансировка загрузки про-

цессоров. Но используемый квазитрехмерный подход затрудняет учет продольных эффектов при критически высокой плотности пучка, когда за короткий промежуток времени пучок деформируется или даже разрушается, а также возникают трудности моделирования таким способом при сравнительно большом угле пересечения пучков.

При этом интерес представляет изучение эффектов на расстоянии порядка дисперсии в поперечном направлении. Сама эта величина является небольшой, за счет фокусировки пучка (hour-glass эффект) его размеры в поперечном направлении сильно увеличиваются. Поэтому необходимо проводить исследования эффектов при больших размерах области на малых расстояниях от ее центра, то есть брать достаточно много и мелких пространственных шагов в поперечном направлении, что также приводит к необходимости использования параллельного алгоритма. Кроме того, временной шаг требуется выбирать не из соображений сходимости, так как он уже достаточно мал, а из соображений устойчивости метода. Таким образом, уменьшение сетки в поперечном направлении ведет не только к увеличению количества действий, связанных с количеством узлов сетки, но и к увеличению количества шагов программы.

Повышение точности расчетов методом части может быть достигнуто увеличением количества узлов пространственной сетки, однако такой способ не всегда возможно применять ввиду ограниченности памяти ЭВМ. Кроме того, для решений методом частиц характерны численные шумы, «самосила» и другие негативные эффекты, связанные с введением пространственной сетки. Одним из способов уменьшения влияния этих эффектов является увеличение количества частиц в ячейке. Кроме того, при увеличении количества узлов по каждому направлению, количество частиц также должно быть увеличено. Поэтому единственным вариантом достижения требуемых количественных характеристик является распараллеливание.

Поэтому для распараллеливания алгоритма для обеих задач выбран метод декомпозиции. Наиболее простая декомпозиция представляет собой разделение области на полосы в соответствующей системе координат. Для наибольшей эффективности вычислений с целью уменьшить число межпроцессорных коммуникаций необходимо модифицировать имеющиеся алгоритмы с учетом специфики решаемых задач [14]. Модификация представляет собой эйлеро-лагранжеву декомпозицию [15].

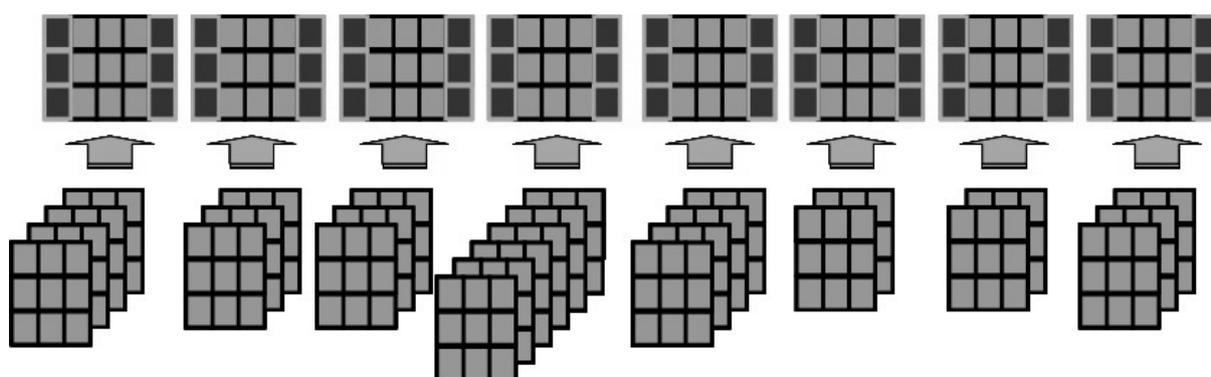


Рис.2. Эйлерово-лагранжева декомпозиция.

С каждой подобластью-полосой связана группа процессоров, частицы в каждой подобласти разделены между всеми процессорами группы, поэтому, чем больше частиц в определенной области – тем больше процессоров в соответствующей группе. Выбором количества процессоров в группе можно добиться необходимого максимального количества частиц в процессоре. Схема распараллеливания приведена на рисунке 2. Таким образом, в случае неравномерного распределения частиц по пространству, на каждую полосу приходится количество процессоров пропорциональное количеству частиц в полосе.

Если же частицы движутся с различным шагом по времени, то частицы распределяются между процессорами полосы не равномерно. Поскольку величина магнитного поля в пределах ячейки существенно не меняется, можно для каждой ячейки ввести поправочный коэффициент, характеризующий величину шага по времени для частиц данной ячейки. Чем больше магнитное поле – тем меньше поправочный коэффициент. Исходя из этих рассуждений, для того, чтобы вычислительная нагрузка на процессоры была сбалансирована, число процессоров N_{pg} в группе оценивается следующим образом:

$$N_{pg} = \sum_{j \in \text{local area}} \frac{N_j}{t_j} N_p \bigg/ \sum_{j \in \text{all area}} \frac{N_j}{t_j}. \quad (10)$$

Здесь N_j — число частиц в ячейке j , t_j — поправочный коэффициент, N_p — общее число процессоров.

На рисунке 3 приведена блок-схема алгоритма. Здесь белым цветом обозначены блоки программы, выполняемые каждым процессором, серым — нулевым процессором, и темно-серым — главными процессорами каждой группы. Индексом g обозначены значения функция, вычисленные каждым процессором группы, которые далее используются главным процессором группы для получения соответствующей суммарной функции. Обозначение `send` описывает пересылку значений каждым процессором группы главному процессору группы, а обозначение `recv` — принятие значений главным процессором группы от всех процессоров его группы, обозначение `bcast` — отправку значений главного процессора всем процессорами группы и соответствующее получение их. В случае ловушки требуются дополнительные процедуры по добавлению/удалению частиц. Этого не требуется в случае встречных пучков, но взамен необходимо вычислять граничные условия на каждом шаге по времени. Поэтому переход к каждому из блоков показан пунктирной линией.

После вычисления начальных условий выполняется цикл по времени до достижения нужного момента времени. В этом цикле требуется решать уравнения Максвелла, и каждая группа решает их только в своей подобласти. Уравнение Максвелла для вычисления электрического поля решается только главными процессорами группы, для этого производится сбор необходимых данных — плотностей и токов, а также обмен значениями на смежных гранях. За счет использования дополнительных узлов и равных значений электрического поля во всех процессорах магнитное поле вычисляется каждым процессором без пересылок. Также группы обмениваются частицами, оказавшимися в соответствующей подобласти. Такое распараллеливание по пространству и по частицам

позволяет решить проблему существенно неравномерного распределения плотности частиц в области, обеспечивая равномерную загрузку процессоров внутри группы.

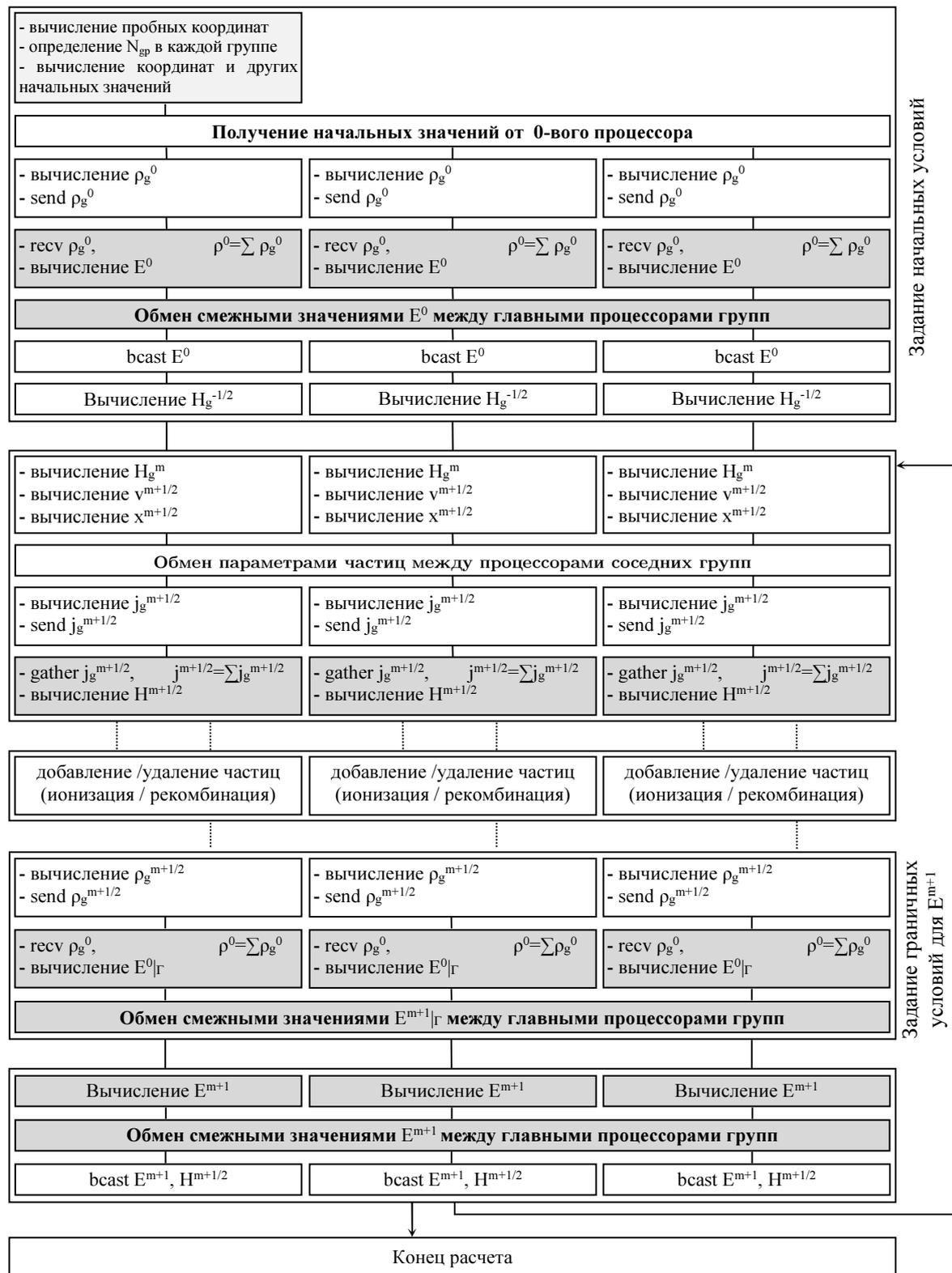


Рис. 3. Блок-схема алгоритма

3. Эффективность параллельного алгоритма

Описанный алгоритм был протестирован на некоторых характерных примерах движения ультрарелятивистского пучка. В частности, предполагается, что пучок моноэнергетических электронов движется вдоль оси z . Плотность частиц распределена по закону Гаусса со следующими параметрами фокусировки в безразмерных величинах $\epsilon_x = \epsilon_y = 5 \cdot 10^{-7}$, $\beta_x = \beta_y = 0,1$, $\sigma_z = 0,1$, где ϵ_x and ϵ_y горизонтальный и вертикальный эмиттансы пучка, β_x, β_y — соответствующие значения бета-функции, σ_z — размер пучка по оси z . Релятивистский фактор $\gamma = 6,85 \cdot 10^3$, заряд пучка $Q = 2,63 \cdot 10^8 e$. Размер области в безразмерных величинах $L_x = L_y = 10^{-2}$, $L_z = 1$. Размер сетки $120 \times 120 \times 120$ узлов, временной шаг 10^{-5} , количество шагов $3 \cdot 10^4$.

Все численные эксперименты проводились на кластере Сибирского Суперкомпьютерного Центра (ИВМиМГ), с 576-ю 4-ядерными процессорами Intel Xeon E5450/E5540/X5670.

На рис. 4 показана зависимость поля E_x от координаты z в плоскости (x,z) при разном количестве частиц $10^5, 10^6, 10^7$ в конечный момент времени в безразмерных величинах.

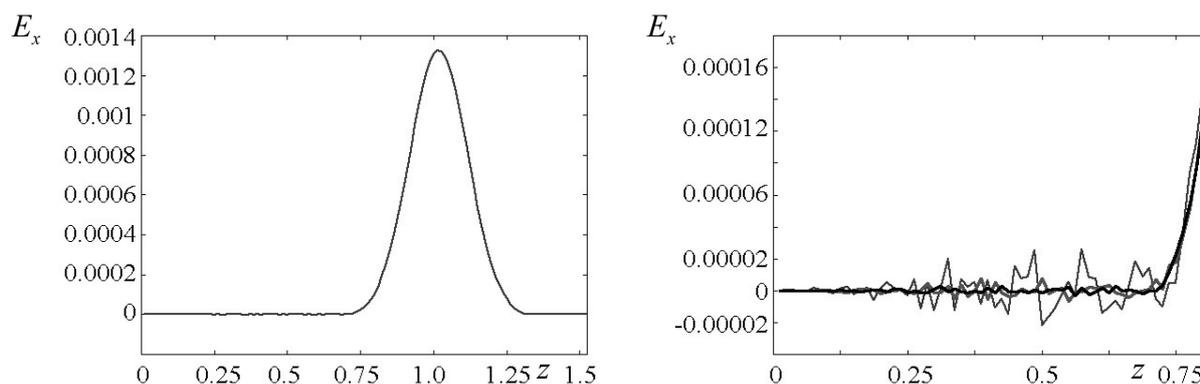


Рис. 4. Зависимость поля E_x от координаты z в плоскости (x,z) во всей области (слева) и в более мелком масштабе (справа).

Из рисунков видно, что при увеличении количества частиц остаточное поле, следующее за пучком, уменьшается, сглаживается. Рисунки также демонстрируют сходимость метода. Если же количество частиц в ячейке недостаточно, то высокие значения шумов в методе частиц приводят к неустойчивостям, затрудняющим исследование физических эффектов.

В табл. 1 представлены время расчета пролета пучка через область и максимальное количество частиц в процессоре, как в начальный момент (предпоследняя колонка), так и в течение всего расчета (последняя колонка). Таблица демонстрирует преимущество добавочных процессоров в группах, соответствующих высокой плотности частиц. Для линейной декомпозиции с количеством процессоров $N_p = N_{pg} = 12$ время вычислений составило 84 с, и 98 с — для $N_p = N_{pg} = 6$, но время расчета намного меньше, когда используется всего 12 процессоров в 6 группах. Таким образом, намного эффективнее

Время расчетов и максимум частиц в процессоре

N_{pg}	N_p	$500\tau, c$	$j_{max}, 10^6, t=0$	$j_{max}, 10^6$
6	6	98	475025	500162
6	10	39	159123	167167
10	10	93	396014	497950
10	12	49	198030	248795
12	12	84	120590	494225
6	12	37	119283	125476

добавить 4–6 процессоров в центральные группы и получить время 37–39 с, чем распределить эти процессоры в линию и получить время 93 с. Этот эффект связан с существенным уменьшением числа частиц ($c 4 \cdot 10^5$ до $1-1,5 \cdot 10^5$) в начальный момент времени, что влечет за собой и уменьшение количества частиц по всему расчету до $1,2-1,8 \cdot 10^5$. Различие между 10 и 12 процессорами в 6 группах невелико вследствие межпроцессорных пересылок внутри группы.

Рисунок 5 демонстрирует зависимость эффективности распараллеливания от количества процессоров N (N_p) для $N_{pg}=5$, сетки $60 \times 60 \times 60$ и числа частиц 10^6 . Время счета 1000 временных шагов составило 300 с. Эффективность распараллеливания рассчитывалась по формуле $Eff = N_0 T_N / T_{N_0}$, где N — количество процессоров, N_0 — начальное количество процессоров. В идеальном случае необходимо сравнивать с монопроцессорным вариантом алгоритма, $N_0=1$, но для проведения такого расчета недостаточно ресурсов памяти ЭВМ. Поэтому мы считали, что эффективность равна 1,0, когда мы используем $N=N_{pg}=5$. T_N и T_{N_0} — время расчетов с соответствующим количеством процессоров.

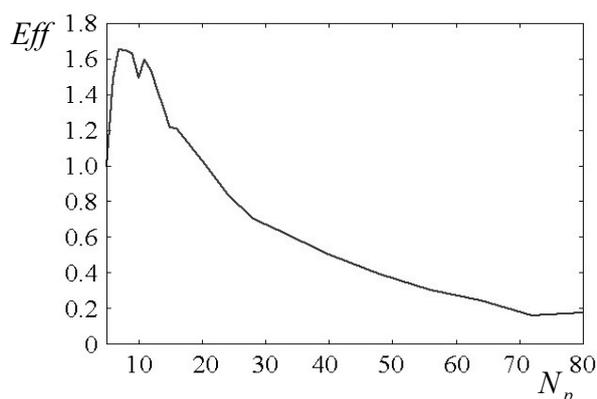


Рис. 5. Эффективность параллелизации $N_p=5$, $J=10^6$, сетка $60 \times 60 \times 60$

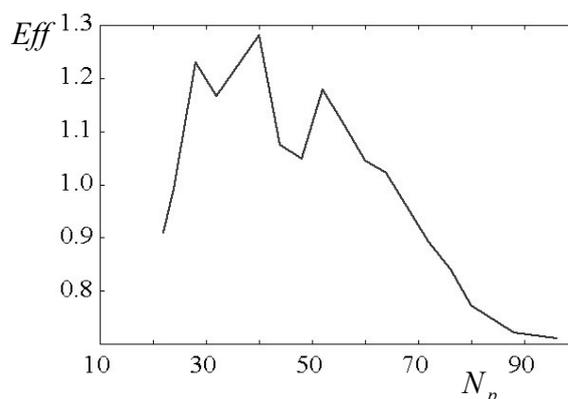


Рис. 6. Эффективность параллелизации $N_p=20$, $J=10^7$, сетка $120 \times 120 \times 60$

Добавление новых процессоров в области высокой плотности существенно увеличивает эффективность метода, когда число процессоров не очень велико (~ 20 в 5 группах),

но далее обмен трехмерными массивами сеточных значений занимает все больше времени, а поэтому эффективность распараллеливания падает.

Аналогичный результат получен и для параметров $N_p = 20$ на сетке $120 \times 120 \times 60$, число частиц 10^7 , время расчета 1000 шагов по времени 67 с (рис. 6.). При этом провалы на графике связаны с несимметричным распределением процессоров в группах, поэтому некоторых из них простаивают. Из численных экспериментов следует, что оптимальное соотношение процессоров N_{pg} и N_p достигается, когда число групп N_{pg} примерно в 7 раз меньше размера сетки вдоль направления распараллеливания, а общее число процессоров N_p больше количества групп N_{pg} примерно в 3–5 раз. В случае большой сетки эффективность падает за счет пересылок копий сеточных значений, тем не менее – это единственный способ проведения расчетов с таким большим количеством частиц и с таким мелким шагом по времени. Т.к. разумный минимум узлов в процессоре N_y равен 4 (2 узла + 2 вспомогательных узла), то параметры расчета ограничены только ресурсами ЭВМ для массивов размером $\sim 4N_x N_z$, соответствующих сеточным значениям подобласти.

В качестве второй задачи рассматривается динамика плазменных электронов в ловушке мишени. Поскольку задача двумерна, объем пересылаемых данных существенно ниже. В этом случае возможно добиться достаточно высокой масштабируемости алгоритма.

Характеристики задачи: температура плазмы 5 эВ, размер области 6,1 см x 1,2 см. Сетка 4096×128 узлов, общее число модельных частиц 5 242 880 000. Расчеты проводились на суперкомпьютере «Ломоносов» с использованием до 8192 процессорных ядер.

В таблице 2 представлено время расчета одного шага (в секундах) при использовании различного количества процессорных ядер, а также полученное при этом ускорение. В связи с большим объемом требуемой оперативной памяти, масштабируемость рассматривается относительно 1024 процессорных ядер.

Таблица 2

Время счета одного шага для различного числа процессоров и полученное ускорение

Общее количество используемых процессорных ядер	Число процессорных ядер на подобласть	Время расчета одного шага, с.	Ускорение по сравнению с 1024 процессорными ядрами
1024	16	1,411	1
2048	32	0,778	1,81
4096	64	0,362	3,87
8192	128	0,183	7,71

На рис. 7 показаны траектории движения плазменных электронов, имеющих энергию 5 эВ, а также траектории движения электронов, стартующих с катода с энергией 150 эВ. Рисунок показывает, что инверсные магнитные пробки на торцах достаточно хорошо удерживают ловушки и границе инверсных пробок. Для количественного описания потерь плазму в ловушке, в то же время присутствуют

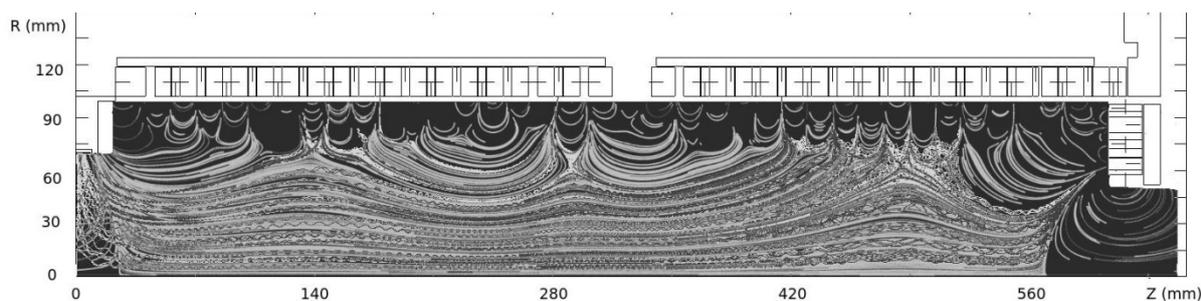


Рис. 7. Траектории движения электронов мишенной плазмы под воздействием магнитного поля

потери плазмы на стенках плазмы в дальнейшем будет учтено влияние рассеяния электронов плазмы.

Заключение

Реализован параллельный алгоритм решения задач динамики плазмы, основанный на модификации эйлерово-лагранжевой декомпозиции в случае неравномерного распределения частиц по пространству и времени. В качестве примеров применения разработанного алгоритма были рассмотрены две задачи - динамики встречных ультрарелятивистских пучков заряженных частиц в ускорителях и динамики плазменных электронов в ловушке с инверсными магнитными пробками и мультипольными магнитными стенками. Для задачи динамики встречных пучков уменьшение остаточного поля, следующего за пучком, с увеличением счетных параметров подтвердило сходимость метода. Для задачи ловушки получены траектории движения плазменных электронов и точки выхода плазмы на стенки ловушки. Применение алгоритма в обоих случаях позволило достичь высокой масштабируемости и увеличить число частиц в расчетах до 10^{10} . Использование большого количества частиц даст возможность проводить расчеты с другими конфигурациями полей и частиц, и достигать большей точности решений.

Работа выполнена при поддержке грантов РФФИ 14-01-00392, 14-01-31220, а также МИП СО РАН №105 и №130.

Литература

1. Вшивков, В. А. Трехмерное моделирование динамики ультрарелятивистских пучков заряженных частиц: особенности вычисления начальных и граничных условий / В.А. Вшивков, М.А. Боронина // Математическое моделирование. — 2012. — Том 24, №2. — С. 67-83.
2. Хокни, Р. Численное моделирование методом частиц. / Хокни Р., Иствуд Дж. — М.: Мир, 1962
3. Березин, Ю.А. Метод частиц в динамике разреженной плазмы. / Березин Ю.А., Вшивков В.А. — Новосибирск, «Наука», 1980.
4. Власов, А.А. Теория многих частиц. / Власов А.А. — М.-Л., ГИТТЛ, 1950, 348 с.

5. Langdon, A.B. Electromagnetic and relativistic plasma simulation models / Langdon A.B, Lasinski B.F. // Meth. Comput. Phys. — 1976. — Vol. 16. — P. 327-366.
6. Villasenor, J. Rigorous charge conservation for local electromagnetic field solver / Villasenor J., Buneman O. // Computer Phys. Comm. — 1992. — Vol. 69. — P. 306-316.
7. Dimov, G.I. Feasible scenario of startup and burnup of fusion plasma in ambipolar D-T reactor / Dimov G.I. // Transactions of Fusion Science and Technology. — 2011. — Vol. 59, No.1T. — P. 208-210.
8. Birdsall, C.K. Particle-in-Cell Charged-Particle Simulation Plus Monte Carlo Collisions With Neutral Atoms, PIC-MCC / Birdsall C.K. // IEEE Trans. Plasma Sci. — 1991. Vol. 19, No. 2. — P. 65-83.
9. C. Rimbault. GUINEA-PIG: A tool for beam-beam effect study // EUROTeV workshop. — 2006. — Vol. Daresbury, 26-27 April.
10. Anderson, E. B. ODYSSEUS: A Dynamic Strong-Strong Beam-Beam Simulation for Storage Rings / E. B. Anderson, T. I. Banks, J. T. Rogers // International Computational Accelerator Physics Conference. — 1998.
11. Bassetti, M. Closed Expression for the Electric Field of a Two-Dimensional Gaussian Charge / M. Bassetti, G. Erskine // CERNISR-ISR-TH/80-06. — 1980.
12. Kabel, A. A Multi-bunch, Three-dimensional, Strong-strong Beam-beam Simulation Code for Parallel Computers / A. Kabel, Y. Cai. // 9th European Particle Accelerator Conference. — 2004.
13. Qiang, Ji. Parallel Strong-Strong/Strong-Weak Simulations of Beam-Beam Interaction in Hadron Accelerators / Ji Qiang, Miguel Furman, Robert D. Ryne, Wolfram Fischer, Tanaji Sen, Meiqin Xiao // AIP Conference Proceedings. — 2003. — Vol. 693. — P. 278-281.
14. Андрианов, А.Н. Подход к параллельной реализации метода частиц в ячейках / Андрианов А.Н., Ефимкин К.Н. // Препринты ИПМ — Москва, 2009 г. — №009, 20с.
15. Берендеев, Е.А. Реализация эффективных параллельных вычислений при моделировании больших задач физики плазмы методом частиц в ячейках / Берендеев Е.А., Ефимова А.А. // Мат. междунар. конф. «Параллельные вычислительные технологии». Новосибирск, 2012. — С. 380-385.

Берендеев Евгений Андреевич, аспирант, ИВМиМГ СО РАН (Новосибирск, Российская Федерация), evgeny.berendeev@gmail.com

Боронина Марина Андреевна, к.ф.-м.н., младший научный сотрудник, Институт вычислительной математики и математической геофизики (Новосибирск, Россия), boronina@parbz.sccc.ru.

Корнеев Владимир Дмитриевич, к.т.н., доцент, старший научный сотрудник, Институт вычислительной математики и математической геофизики (Новосибирск, Россия), korneev@sccc.sccc.ru.

Поступила в редакцию 26 февраля 2014 г.

PARALLEL ALGORITHM FOR SOLUTION OF PROBLEMS OF CHARGED PARTICLE DYNAMICS BY THE USE OF LOAD BALANCE

E.A. Berendeev, Institute of Computational Mathematics and Mathematical Geophysics
SB RAS (Novosibirsk, Russian Federation)

M.A. Boronina, Institute of Computational Mathematics and Mathematical Geophysics
SB RAS (Novosibirsk, Russian Federation)

V.D. Korneev, Institute of Computational Mathematics and Mathematical Geophysics
SB RAS (Novosibirsk, Russian Federation)

We present new solution methods for dynamics of counter charged beams in accelerators and plasma electrons dynamics in a trap with inverse magnetic mirrors and multipole magnetic walls. The models are based on the particle-in-cell method. These problems require extremely large computations and can be solved by the use of powerful supercomputers only. A modification of the euler-lagrangian decomposition is implemented in order to achieve full and balanced load of computational nodes in case of highly nonuniform particle distribution in space and time.

Keywords: *particle-in-cell method, parallel algorithms, load balance, plasma physics, counter beams, particle accelerators.*

References

1. Vshivkov V.A., Boronina M.A. Trekhmernoe modelirovanie dinamiki ultrarelyativistskikh puchkov zaryazhennykh chastic: osobennosti vychisleniya nachalnykh i granichnykh usloviy. *Mathematical Models and Computer Simulations*, 2012, vol. 24, no 2, pp. 67–83.
2. Hockney R.W., Eastwood J.W. *Chislennoe modelirovanie metodom chastits*. Moscow: Mir, 1962.
3. Berezin Ju.A., Vshivkov V.A. *Metod chastits v dinamike razrezhennoj plazmy*. Novosibirsk, «Nauka», 1980.
4. Vlasov A.A. *Teoria mnogikh chastits* — Moscow-Leningrad., GITTL, 1950, 348 p.
5. Langdon A.B, Lasinski B.F. Electromagnetic and relativistic plasma simulation models. *Meth. Comput. Phys.*, 1976, vol. 16, pp. 327–366.
6. Villasenor J., Buneman O. Rigorous charge conservation for local electromagnetic field solver. *Computer Phys. Comm.*, 1992, vol. 69, pp. 306–316.
7. Dimov G.I. Feasible scenario of startup and burnup of fusion plasma in ambipolar D-T reactor. *Transactions of Fusion Science and Technology*, 2011, vol. 59, no. 1T, pp. 208–210.
8. Birdsall C.K. Particle-in-Cell Charged-Particle Simulation Plus Monte Carlo Collisions With Neutral Atoms, PIC-MCC. *IEEE Trans. Plasma Sci.*, 1991, vol. 19, no. 2, pp. 65–83.

9. Rimbault. C. GUINEA-PIG: A tool for beam-beam effect study. EUROTeV workshop, 2006, vol. Daresbury, 26–27 April.
10. Anderson E.B., Banks T.I., Rogers J.T. ODYSSEUS: A Dynamic Strong-Strong Beam-Beam Simulation for Storage Rings. International Computational Accelerator Physics Conference, 1998.
11. Bassetti M., Erskine G. Closed Expression for the Electric Field of a Two-Dimensional Gaussian Charge. CERNISR-ISR-TH/80-06, 1980.
12. Kabel A., Cai Y. A Multi-bunch, Three-dimensional, Strong-strong Beam-beam Simulation Code for Parallel Computers, 9th European Particle Accelerator Conference, 2004.
13. Qiang J., Furman M., Ryne R.D., Fischer W., Sen T., Xiao M. Parallel Strong-Strong/Strong-Weak Simulations of Beam-Beam Interaction in Hadron Accelerators. AIP Conference Proceedings, 2003, vol. 693, pp. 278–281.
14. Andrianov A.N., Efimkin K.N. Podkhod k parallelnoj realizatsii metoda chastits v yachejkakh. Preprinty IPM, Moscow, 2009, no. 009, 20 p.
15. Berendeev E.A., Efimova A.A. Realizatsija effektivnykh parallelnykh vychislenij pri modelirovanii bolshikh zadach fiziki plazmy metodom chastits v yachejkakh. Materialy mezhdunarodnoy konferentsii “Parallelnye vychislitelnye tekhnologii”. Novosibirsk, 2012, pp. 380–385.

Received 26 February 2014

Краткие сообщения

УДК 004.042

МЕТОД РАСПРЕДЕЛЕННОГО ОБНАРУЖЕНИЯ ИЗМЕНЕНИЯ КОНЦЕПЦИИ

А.А. Волков, Л. Бюх, А. Андряк

Представлен метод распределенного обнаружения изменения концепции для алгоритмов интеллектуального анализа данных. Под изменением концепции понимается любое непредсказуемое изменение входных данных алгоритма. Предложена реализация метода с использованием технологии распределенных вычислений MapReduce. Разработанный алгоритм предназначен для обнаружения изменения концепции в потоке входных данных в режиме реального времени. С целью обеспечения итеративного поведения фаз Map и Reduce разработан специальный MapReduce-фреймворк и осуществлена его программная реализация. Использование алгоритма позволит автоматически обнаруживать изменение входных данных, требующее изменение параметров используемой модели и переключение на использование новой модели в режиме реального времени.

Ключевые слова: изменение концепции, интеллектуальный анализ данных, распределенные вычисления, итеративный MapReduce.

Введение

Благодаря современному развитию технологий мы можем накапливать огромные объемы данных в любых областях науки. Для анализа этих данных используются различные модели и алгоритмы. Однако реальные данные редко имеют постоянное распределение, и, следовательно, не могут описываться одной моделью на протяжении длительного периода времени. В связи с этим возникает необходимость каким-либо образом обнаруживать скачкообразные изменения во входных данных с целью своевременного изменения параметров модели.

В результате изменения входных данных использование текущей модели приводит к неверным результатам и становится неэффективным, что приводит к необходимости изменить параметры используемой модели. Такое изменение входных данных называют изменением концепции (concept drift) [1, 2, 5, 6]. При этом под концепцией понимается любая сущность реального мира, которая подвергается интеллектуальному анализу.

Возможность автоматического обнаружения изменения концепции во время анализа данных позволит существенно ускорить процесс получения результата и значительно снизит ошибки анализа данных.

В работах [2, 5, 6] предложены последовательные алгоритмы обнаружения изменения концепции, авторы статьи [1] предлагают распределенный и параллельный подход. Основным средством для отслеживания возможного изменения концепции, авторы перечисленных работ используют изменение значения ошибки. Способ получения значений ошибки зависит от конкретной области интеллектуального анализа данных.

Во всех рассмотренных подходах для получения новой модели используются элементы входного потока данных, начиная с элемента, при котором впервые было зафиксировано состояние возможного изменения концепции (warning level). Переход на новую модель осуществляется в точке изменения концепции, когда достигнут заданный уро-

вень ошибки (drift level). Отличием [1] является то, что настройка новой модели осуществляется параллельно с использованием основной модели.

В данной работе предлагается усовершенствованный вариант алгоритма распределенного обнаружения изменения концепции, описанного в [1]. В качестве инструмента, позволяющего реализовать параллельную, распределенную и устойчивую к сбоям обработку больших массивов данных, использована парадигма программирования MapReduce с использованием расширений, предложенных в [3], обеспечивающих возможность обработки потоковых данных в инкрементном режиме.

Предложенный алгоритм относится к классу итеративных MapReduce алгоритмов. Для учета специфики алгоритма была разработана специальная программная структура (фреймворк).

В разделе 1 описана реализация предложенного усовершенствованного алгоритма распределенного обнаружения изменения концепции. В разделе 2 представлено описание MapReduce-фреймворка, обеспечивающего исполнение разработанного алгоритма. В заключении изложены возможные пути дальнейшего развития предложенного алгоритма.

1. Алгоритм распределенного обнаружения изменения концепции

1.1. Причины необходимости доработки предыдущей реализации

В статье [1] подробно описывается алгоритм распределенного обнаружения изменения концепции, реализованный с использованием парадигмы программирования MapReduce. Однако практические испытания данного алгоритма указали на ряд неучтенных факторов, снижающие его эффективность.

Во-первых, Map-элементы получают сообщения об изменении концепции (warning или drift) со значительным запозданием. То есть Map-элемент может обработать и отправить Reduce-элементу сотни значений после того момента времени, когда произошло изменение концепции. Чтобы не потерять точность вычислений, Map-элементу необходимо повторно обработать элементы входного потока, начиная с элемента, соответствующего последнему сообщению об изменении концепции.

Данная проблема в [1] решалась путем синхронизации Map и Reduce элементов, однако синхронизация, как известно, приводит к снижению эффективности распределенных алгоритмов.

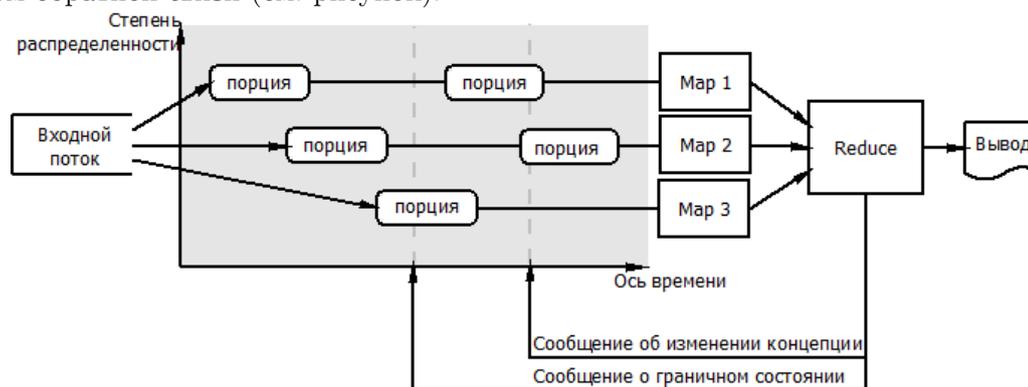
Во-вторых, в случае если Map-элемент начинает повторно обрабатывать элементы входного потока, но с использованием новой модели, Reduce-элемент должен иметь возможность фильтровать значения, полученные с использованием разных моделей, и выбирать только актуальные результаты.

Кроме того, для повышения наглядности и прозрачности алгоритма, был введен ряд абстрактных структур, которые более подробно рассмотрены далее.

1.2. Общая схема алгоритма

Входящий поток данных делится на порции и обрабатывается Map-элементами. Пары <ключ, значение>, произведенные всеми Map-элементами, группируются по ключу и направляются в виде пары <ключ, массив значений> в адрес Reduce-элементов. Reduce-элементы, независимо друг от друга, обрабатывают эти пары и выдают окончательный результат. Вместе с тем, Reduce-элементы могут посылать Map-элементам со-

общения о нормальном, граничном состоянии, или состоянии изменения концепции посредством обратной связи (см. рисунок).



Общая схема алгоритма обнаружения изменения концепции

Таким образом, процесс обучения модели включает следующие стадии:

1. Разделение входного потока на «порции», которые раздаются Мар-элементам по очереди. Размер порции является одним из параметров алгоритма. Он не должен быть маленьким, чтобы снизить расходы на выделение и освобождение памяти, но и не должен быть большим, чтобы издержки на передачу данных отдельным рабочим узлам не превышали время обработки этих данных рабочим узлом.
2. Каждый Мар-элемент строит свою собственную модель, основываясь только на своих входных данных.
3. Reduce-элемент обобщает результаты, поступившие от всех Мар-элементов, и посылает одновременно всем Мар-элементам одно из трех сообщений о состоянии текущей модели (нормальное, граничное, изменение концепции). Механизм обратной связи предоставляется фреймворком.

Важной особенностью данного алгоритма является то, что обработка Мар-элементом входных данных осуществляется независимо от работы Reduce-элемента, которая включает в себя принятие решения об изменении концепции. То есть, в отличие от распространенного подхода, при котором каждая итерация алгоритма представляет собой последовательность фаз Мар и Reduce, в предложенном алгоритме итерации происходят в Мар и Reduce фазах параллельно.

Решение об изменении концепции Reduce-элемент может принять, лишь обработав последовательность исторических данных. То есть Мар-элемент получит сообщение о том, что изменение концепции имело место в определенный момент в прошлом. В связи с этим, для получения корректного результата, Мар-элементу необходимо обработать данные повторно, начиная с того момента, когда изменение концепции имело место.

Поскольку может возникнуть ситуация, что Reduce-элемент получит результаты обработки одного и того же входного значения, но с использованием разных моделей, независимо друг от друга Reduce и Мар элементы хранят номер текущей, актуальной модели. После рассылки сообщения об изменении концепции Reduce-элемент игнорирует входные значения, обработанные предыдущими моделями.

1.3. Мар-элемент

Каждый Мар-элемент имеет кэш, в который он записывает входные данные для того, чтобы обеспечить возможность их повторной обработки.

Для получения сообщений от Reduce-элемента о нормальном, граничном состоянии или состоянии изменения концепции Map-элемент имеет почтовый ящик. Проверка сообщений и обработка входных данных осуществляются параллельно.

Если Map-элемент получает сообщение о граничном состоянии после сообщения о нормальном состоянии, он продолжает использовать текущую модель, но должен начать обучение новой модели в фоновом режиме, то есть одновременно с использованием текущей модели. Для обучения он использует кэшированные значения входного потока, начиная с позиции, в которой Reduce-элементом зафиксировано граничное состояние. При этом кэш старше последнего сообщения о граничном состоянии может быть очищен. В случае получения сообщения о нормальном состоянии после сообщения о граничном состоянии, модель, обучаемая в фоновом режиме, сбрасывается.

При поступлении сообщения об изменении концепции, Map-элемент сохраняет действующую модель и приступает к использованию модели, обучаемой в фоновом режиме с позиции, в которой Reduce-элемент зафиксировал изменение концепции. При этом порядковый номер действующего классификатора (модели) увеличивается на единицу.

Выходные значения Map-элемента имеют обязательную пометку с номером модели, используемой для его вычисления, а также привязку ко временной оси.

Позиции нужны для того, чтобы Reduce-элементом обрабатывал данные, поступающих от всех Map-элементов, в хронологическом порядке. В противном случае результаты обнаружения изменения концепции будут некорректными.

Номер классификатора используется между Map- и Reduce-элементами как средство идентификации того, что они используют наиболее актуальную модель. Поскольку сообщения от Map- к Reduce-элементам поступают асинхронно, часто возникает ситуация, когда часть Map-элементов системы продолжают отправлять значения, полученные устаревшей моделью. Такие сообщения должны быть проигнорированы Reduce-элементами.

1.4. Reduce-элемент

Поскольку обнаружение явления изменения концепции зависит от момента времени, необходимо обеспечить, чтобы Reduce-элемент обрабатывал поступившие к нему от Map-элементов значения не в порядке их поступления, а в соответствии с их привязкой к временной оси.

Для этого Reduce-элемент использует специальную структуру – скользящее окно. Все значения, поступающие от Map-элементов, имеют привязку ко времени. В соответствии с этой привязкой, входные значения Reduce-элемента помещаются в скользящее окно. Поскольку значения от Map-элементов поступают асинхронно, скользящее окно может содержать пустые места (пропуски). Обработка полученных значений может производиться только над непрерывной временной последовательностью значений с начала (левой границы) скользящего окна. Обработанные значения удаляются, начало скользящего окна смещается до временной метки последнего обработанного значения.

Для исключения возможности попадания в скользящее окно неактуальных значений, полученных с помощью устаревших моделей, Reduce-элемент производит проверку всех поступающих значений. Для этого все входные данные имеют пометку Map-элемента с номером модели, используемой для получения данного результата. Reduce-элемент ведет собственный учет текущей модели, и в случае несовпадения этих номеров – игнорирует входное значение.

В случае обнаружения Reduce-элементом состояния изменения концепции происходит полная очистка скользящего окна.

Reduce-элемент, по результатам обработки последовательности значений из скользящего окна, отправляет Map-элементам сообщение о нормальном, граничном состоянии или состоянии изменения концепции посредством механизма обратной связи. Данное сообщение рассылается фреймворком в ящики входящих сообщений всех Map-элементов.

В случае завершения входного потока, либо достижения Reduce-элементом условия останова, Reduce-элемент генерирует сообщение об останове работы программы и рассылает его Map-элементам.

2. Фреймворк для итеративного MapReduce

Большинство программных структур для алгоритмов MapReduce не предназначено для реализации итеративных алгоритмов [4]. Проблемы, в первую очередь, связаны с необходимостью чтения и записи данных в начале и окончании каждой итерации соответственно.

Предложенный алгоритм имеет иной принцип организации итеративного поведения: итерации осуществляются внутри Map- и Reduce-элементов независимо друг от друга. Для обеспечения такого поведения был разработан специальный MapReduce-фреймворк.

Разработанный фреймворк обеспечивает Hadoop-совместимый интерфейс программирования приложений (API), а также предоставляет дополнительные функциональные возможности для реализации итеративного поведения MapReduce алгоритмов.

Основными особенностями разработанного фреймворка являются:

1. Возможность обратной связи между Reduce-элементами и Map-элементами;
2. Возможность получения предварительного результата;
3. Реализация итеративного поведения алгоритмов MapReduce;
4. Инкрементная обработка больших объемов данных [3].

Рассмотрим данные особенности более подробно.

2.1. Наличие обратной связи

Предлагаемый фреймворк предусматривает два типа обратной связи:

- Broadcast;
- Restream.

Обратная связь Broadcast – это сигнал, поступающий от Reduce-элемента всем Map-элементам по типу рассылки электронной почты. У каждого Map-элемента есть свой «ящик», который он периодически проверяет на наличие новых сообщений. Данный тип обратной связи предназначен для посылки различных служебных сообщений: изменение глобальных параметров, необходимость очистки памяти (кэша) входных данных, либо преждевременный останов программы.

При обратной связи Restream пересылаемый сигнал представляет собой обычную пару <ключ, значение>, которая попадает во входной поток и обрабатывается как обычный вход. Данную пару получит только один Map-элемент, причем конкретный номер Map-элемента, который будет ее обрабатывать, зависит от диспетчера Map-задач. Данный тип обратной связи применяется в случае, когда нужно повторно обработать какие-то входные данные (если предварительный результат не удовлетворяет определенным условиям).

2.2. Использование предварительных результатов

Разработанный фреймворк позволяет получить предварительные результаты вычислений [3]. Использование предварительных результатов позволяет оценить качество модели без необходимости обрабатывать все входные данные полностью. Если качество предварительного результата не удовлетворяет заданным требованиям, фреймворк позволяет осуществить преждевременный останов программы.

Эти возможности позволяют уменьшить время до получения решения. Пользователь может получить знание о том, способно ли некоторое изменение параметров задачи привести к улучшению окончательного решения без необходимости решать задачу до выполнения условия останова.

2.3. Реализация итеративного поведения

Каждый Map-элемент существует постоянно на протяжении всего времени работы алгоритма, и все данные, необходимые для работы Map-элемента, хранятся в оперативной памяти. В результате этого не происходит лишних операций чтения и записи при каждой итерации. Чтение входных данных и их обработка осуществляются параллельно.

Таким образом, представленный фреймворк избавлен от накладных расходов, связанных с чтением и записью данных на каждой итерации.

Каждый Map-элемент частями отправляет результаты своих вычислений во входной буфер Reduce-элемента, и продолжает вычисления над другими данными. После того, как Reduce-элемент выполнит необходимые действия над данными, находящимися в его входном буфере – он выдает предварительный результат и, при необходимости, посредством обратных связей, посылает корректирующие сообщения Map-элементам.

Заключение

В сообщении описана реализация алгоритма распределенного обнаружения изменения концепции с использованием технологии программирования MapReduce. Приведено описание разработанной программной структуры, предоставляющей набор классов и интерфейсов для реализации итеративных алгоритмов MapReduce.

Для обнаружения изменения концепции используется изменение значения ошибки алгоритма анализа данных. Алгоритм обладает высокой степенью масштабирования, и может применяться для обработки очень больших данных.

В дальнейшем работа будет продолжена с целью расширения возможных областей применения данного алгоритма, а также унификации формата входных данных.

Исследование выполнено в рамках реализации программы развития ПНР-5 ФГБОУ ВПО «ЮУрГУ» (НИУ) на 2009–2019 гг.

Литература

1. Andrzejak, A. Parallel Concept Drift Detection with Online Map-Reduce / A. Andrzejak, J.B. Gomes // International Workshop on Knowledge Discovery (KDCLOUD-2012). Dec. 2012. — P. 402–407.

2. Baena-Garcia, M. Early drift detection method / M. Baena-Garcia, J. Campo-Avila, R. Fidalgo, A. Bifet, R. Gavaldá, R. Morales-Bueno // The 4th International Workshop on Knowledge Discovery from Data Streams. Sep. 2006. — P. 77–86.
3. Bose, J.-H. Beyond online aggregation: Parallel and incremental data mining with online mapreduce / J.-H. Bose, A. Andrzejak, M. Hogqvist // ACM Workshop on Massive Data Analytics over the Cloud (MDAC 2010). Apr. 2010.
4. Doukeridis, C. A survey of large-scale analytical query processing in MapReduce / C. Doukeridis, K. Nørnvåg // The VLDB Journal. 2013. — P. 1–26.
5. Gama, J. Learning with drift detection / J. Gama, P. Medas, G. Castillo, P. Rodrigues // Advances in Artificial Intelligence. Nov. 2004. — Vol. 3171. — P. 286–295.
6. Sobhani, P. New Drift Detection Method for Data Streams / P. Sobhani, H. Beigy // Adaptive and Intelligent Systems. Sep. 2011. — Vol. 6943. — P. 88–97.

Волков Антон Александрович, магистрант факультета Вычислительной математики и информатики, Южно-Уральский государственный университет (Челябинск, Российская Федерация), potnavol@gmail.com.

Люц Бюх, Институт информатики, Гейдельбергский университет (Германия), lutz.buech@informatik.uni-heidelberg.de.

Артур Андрияк, доктор., профессор Института информатики, Гейдельбергский университет (Германия), artur.andrzejak@informatik.uni-heidelberg.de.

Поступила в редакцию 2 октября 2013 г.

*Bulletin of the South Ural State University
Series “Computational Mathematics and Software Engineering”
2014, vol. 3, no. 1, pp. 113–120*

A METHOD FOR DISTRIBUTED CONCEPT DRIFT DETECTION

A.A. Volkov, South Ural State University (Chelyabinsk, Russian Federation),

L. Büch, Heidelberg University (Heidelberg, Germany)

A. Andrzejak, Heidelberg University (Heidelberg, Germany)

The paper introduces a method for distributed concept drift detection for data mining algorithms. Concept drift is understood as any unpredictable alteration in input data. There is an algorithm implementation proposed, based on MapReduce distributed computing technology. Proposed algorithm meant for concept drift detection in streaming data in online fashion. In order to provide iterative Map and Reduce phases a MapReduce framework is introduced. The algorithm is able to automatically detect input data alteration, which demands model parameters change and switching a new model online.

Keywords: concept drift, data mining, distributed computations, iterative MapReduce.

References

1. Andrzejak A., Gomes J.B. Parallel Concept Drift Detection with Online Map-Reduce // International Workshop on Knowledge Discovery (KDCloud-2012). Dec. 2012. P. 402–407.
2. Baena-Garcia M., Campo-Avila J., Fidalgo R., Bifet A., Gavaldà R., Morales-Bueno R. Early drift detection method // The 4th Int. Workshop on Knowledge Discovery from Data Streams. Sep. 2006. P. 77–86.
3. Bose J.-H., Andrzejak A., Hogqvist M. Beyond online aggregation: Parallel and incremental data mining with online mapreduce // ACM Workshop on Massive Data Analytics over the Cloud (MDAC-2010). Apr. 2010.
4. Doulkeridis C., Nørnvåg K. A survey of large-scale analytical query processing in MapReduce // The VLDB Journal. 2013. P. 1–26.
5. Gama J., Medas P., Castillo G., Rodrigues P. Learning with drift detection // Advances in Artificial Intelligence. Nov. 2004. Vol. 3171. P. 286–295.
6. Sobhani P., Beigy H. New Drift Detection Method for Data Streams // Adaptive and Intelligent Systems. Sep. 2011. Vol. 6943. P. 88–97.

Received 2 October 2013

СВЕДЕНИЯ ОБ ИЗДАНИИ

Серия основана в 2012 году.

Свидетельство о регистрации ПИ ФС77-26455 выдано 13 декабря 2006 г. Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия.

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).