ОТОБРАЖЕНИЯ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ НА СУПЕРЭВМ ЭКЗАФЛОПСНОЙ ПРОИЗВОДИТЕЛЬНОСТИ С РАЗЛИЧНЫМИ АРХИТЕКТУРАМИ НА ОСНОВЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ.

Б.М. Глинский, М.А. Марченко, Б.Г. Михайленко, А.С. Родионов, И.Г. Черных

REPRESENTATION OF PARALLEL ALGORITHMS ON DIFFERENT ARCHITECTURES OF EXAFLOPS SUPERCOMPUTERS BASED ON SIMULATION MODELING

B.M. Glinskiy, M.A. Marchenko, B.G. Michailenko, A.S. Rodionov, I.G. Chernykh

Целью работы является исследование возможности отображения параллельных алгоритмов на архитектуру суперЭВМ экзафлопсной производительности с использованием метода имитационного моделирования. Приводятся примеры отображения алгоритмов для решения конкретной задачи динамики разреженного газа по методу прямого статистического моделирования, связанному с моделированием реализаций ансамбля тестовых частиц и сеточного метода для решения задачи распространения сейсмических волн в изотропно-неоднородной упругой среде.

The aim scope of the research is a possibility of representation of parallel algorithms on different architectures of exaflops supercomputers based on simulation modeling. There are some examples of algorithms presented that based on low density gas dynamics modeling task and seismic waves propagation in isotropic heterogeneous elastic waves problem.

Агентно-ориентированная система; *имитационное моделирование, масштабируемые параллельные алгоритмы.*

Agent oriented system, simulation modeling, scalable parallel algorithms

Проблема исследования свойств масштабируемости параллельных алгоритмов при их реализации на будущих суперЭВМ экзафлопсной производительности выходит за уровень технологических задач и требует научно-исследовательского подхода к ее решению. Вычислительные алгоритмы, как правило, являются более консервативными по сравнению с развитием средств вычислительной техники. Оценить поведение алгоритмов, разработать модифицированные схемы вычислений можно уже сейчас путем реализации их на имитационной модели, отображающей тысячи и миллионы вычислительных ядер. Имитационная модель позволяет выявить узкие места в алгоритмах, понять, как нужно модифицировать алгоритм, какие параметры необходимо настраивать при его масштабировании на большое количество ядер.

Задача моделирования масштабируемых алгоритмов не является новой, ею занимаются многие группы исследователей во всём мире. Из зарубежных выделим исследования, проводимые в США (Университет Урбана-Шампань, Иллинойс). Одним из основных проектов этого коллектива является проект BigSim (http://charm.cs.uiuc.edu/research/bigsim, руководитель проекта Kale Laxmikant). Проект направлен на создание имитационного окружения, позволяющего разработку, тестирование и настройку посредством моделирования ЭВМ будущих

поколений, одновременно позволяя разработчикам ЭВМ улучшать их проектные решения с учётом специального набора приложений [1].

Из отечественных выделим исследования, проводимые в Институте системного программирования РАН (г. Москва) под руководством академика В.П. Иванникова [2,3]. Этим коллективом разработана модель параллельной программы, которая может эффективно интерпретироваться на инструментальном компьютере, обеспечивая возможность достаточно точного предсказания времени реального выполнения параллельной программы на заданном параллельном вычислительном комплексе. Модель разработана для параллельных программ с явным обменом сообщениями, написанных на языке Java с обращениями к библиотеке MPI, и включена в состав среды ParJava. Модель получается преобразованием дерева управления программы, которое для Java-программ может быть построено путем модификации абстрактного синтаксического дерева. Для моделирования коммуникационных функций используется модель LogGP, что позволяет учитывать специфику распределенной вычислительной системы. Предсказание времени счёта отдельных участков параллельной программы производится с учётом затрат, связанных с управлением MPI, т.е. производится корректировка модельных часов с учётом средней доли процессорного времени, которую занимает нить RTS (Run Time System). Таким образом, проект ParJava, с одной стороны, позволяет решать широкий круг задач по оценке эффективности исполнения параллельных программ на перспективных вычислительных системах, но, с другой стороны, привязан к конкретному языку программирования, что существенно сужает его возможности. Стоит отметить, что оба рассмотренных проекта не учитывают, по крайней мере явно, вопросы отказоустойчивости при исполнении больших программ, в то время как использование в вычислениях одновременно десятков и сотен тысяч, а для отдельных задач и миллионов вычислительных ядер не может их не поставить.

В ИВМиМГ СО РАН развивается мультиагентный подход, который органично подходит для задачи имитации вычислений. В качестве атомарной, независимой частицы в модели вычислений выбран вычислительный узел и исполняемый на нем код алгоритма. Каждый функциональный агент эмулирует поведение вычислительного узла кластера, и программу вычислений, работающую на этом узле. Вычисления представляются в виде набора примитивных операций (вычисление на ядре; запись/чтение данных в память; парный обмен данными; синхронизация данных между вычислителями) и временных характеристик каждой операции [4] .

Разработан пакет AGNES (AGent NEtwork Simulator), который базируется на Java Agent Development Framework (JADE). JADE – это мощный инструмент для создания мультиагентных систем на JAVA, и он состоит из 3-х частей: среда исполнения агентов; библиотека базовых классов, необходимых для разработки агентной системы; набор утилит, позволяющих наблюдать и администрировать МАС (мультиагентная система). Для моделирования больших вычислений важно, что JADE – это FIPA-совместимая, распределенная агентная платформа, которая может использовать один или несколько компьютеров (узлов сети), на каждом из которых должна работать только одна виртуальная JAVA машина.

AGNES использует преимущества, предоставляемые JADE, и расширяет мультиагентную систему до системы моделирования [5]. AGNES состоит из двух типов агентов:

– Управляющие агенты (УА), которые создают среду моделирования.

– Функциональные агенты (ФА), которые образуют модель, работающую в среде моделирования.

Приложение AGNES – это распределенная МАС, называемая платформой. Платформа AGNES состоит из системы контейнеров, распределенных в сети. Обычно на каждом хосте находится по одному контейнеру (но при необходимости их может быть несколько). Агенты существуют внутри контейнеров.

Достоинства пакета AGNES:

-отказоустойчивость;

-сбалансированное распределение нагрузки;

-наличие проблемно-ориентированных библиотек агентов;

-возможность динамического изменения модели в ходе эксперимента.

В настоящее время нет определенного мнения по архитектуре ЭВМ экзафлопсной производительности. В качестве одного из возможных решений предполагаем экстенсивное развитие инструментального вычислительного кластера НКС-30Т+GPU ЦКП ССКЦ СО РАН (многократное увеличение количества существующих ядер) [6]. Тем самым делается оценка производительности «снизу», поскольку естественно ожидать повышения характеристик ядер и интерконнекторов ЭВМ экзафлопсной производительности по сравнению с существующими.

Модель программы представляется взвешенным графом переходов между блоками программы с указанием параллельных ветвей. Временные задержки в блоках определяются на основе измерений, производимых в тестовых прогонах реальных программ на НКС-30Т+GPU. Прогоны реальных программ на конфигурациях с более чем 30 000 ядер позволяют надеяться на учёт в измеренных задержках эффектов от системной составляющей.

Рассмотрим примеры реализации отображения алгоритма на архитектуру экзафлопсной ЭВМ при указанных выше предположениях.

Первая задача связана с изучением возможности масштабирования распределенного статистического моделирования на большое число вычислительных ядер. Это задачи часто требуют моделирования экстремально большого количества независимых реализаций [7]. К числу таких проблем, например, относятся задачи прямого статистического моделирования (ПСМ) течений разреженного газа с учетом химических реакций, задачи переноса излучения и теории дисперсных систем.

Имитационное моделирование проводилось с использованием мультиагентной системы AGNES. Для имитации вычислений статистических методов созданы два класса функциональных агентов.

– DataAgregator: ядро-«сборщик», собирает информацию об вычислениях, обрабатывает и агрегирует её. Возможно иерархическое построение «сборщиков», которые на нижнем уровне обрабатывают данные непосредсвенно вычислителей, а затем передают их вышестоящему агенту DataAgregator. На вершине этой пирамиды всегда стоит одно главное ядро-«сборщик», подготавливающее итоговые данные обо всех вычислениях и сохраняющее их на жесткий диск.

– MonteCarlo: агент, имитирующий расчет статистических методов, ядро-«вычислитель». Каждый агент проводит независимые вычисления согласно схеме вычислений и взаимодействует только с соответствующим DataAgregator. Основными характеристиками агента являются временные и статистические свойства, оценки которых получены на основе реальных вычислений.

В результате работы модели собираются следующие отчеты:

Набор времен, потраченных на каждую итерацию вычислений каждым агентом. Эти времена позволяют получить статистические характеристики протекающих в модели вычислений, для оценки правдоподобия модели.

– Информация о количестве итераций вычислений, совершенных каждым агентом MonteCarlo. При помощи данной статистики можно, например, отследить, как влияет количество вычислителей на скорость расчетов.

– Информация об интенсивности получения данных агентами DataAgregator от вычислителей, либо нижестоящих DataAgregator, в данном случае регистрируется количество полученных за равные промежутки времени пакетов.

– Исходные данные для имитационного моделирования получены с использованием библиотеки PARMONC, предназначенной для использования на современных суперкомпьютерах тера- и петафлопсного уровня [8].

На рис. 1 представлена схема вычислений для имитационной модели.

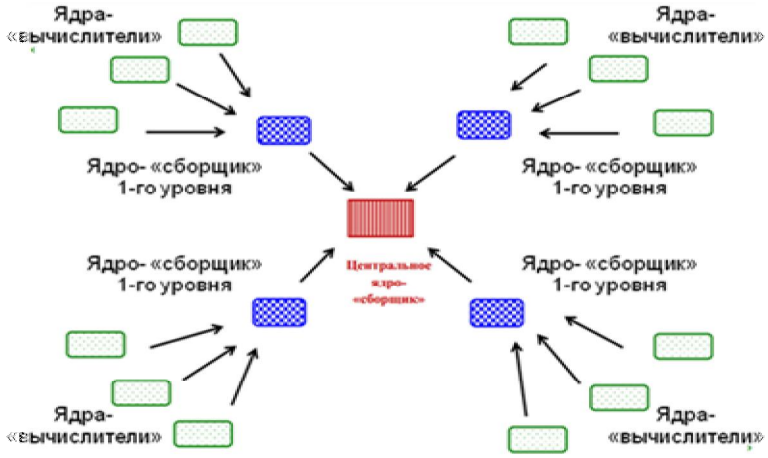

Рис. 1. Имитация работы суперкомпьютера, метод Монте-Карло.

Рассматривались два варианта организации обмена данными с главным ядром-«сборщиком»: одноуровневый и двухуровневый. В двухуровневом варианте ядра-«вычислители» были поделены на N равных частей (N = 10, 20, 100), для каждой из которых данные с ядер-«вычислителей» сначала отправлялись на свое выделенное промежуточное ядро-«сборщик». В свою очередь, N промежуточных ядер-«сборщиков» отправляли данные на главное ядро-«сборщик». В одноуровневом варианте (будем считать, что число промежуточных ядер-«сборщиков» равно нулю: N = 0) данные с ядер-«вычислителей» непосредственно отправлялись на главное ядро-«сборщик».

Ускорение от распараллеливания при расчётах на M ядрах определим так:

$$S_L(M) = T_L(M_{min})/T_L(M)$$

где $T_L(M)$ – машинное время на центральном ядре- «сборщике», затраченное на моделирование и сохранение выборочных средних для L реализаций случайной оценки; M_{min} – наименьшее число ядер, использованных при расчётах.

Приведём сравнение ускорения для имитационной модели с теоретической оценкой, которая в предложении о пренебрежимо малом времени на обмен данными даёт

$$S_L(M) = M/M_{min}$$

Приведем некоторые результаты по оценке масштабируемости, полученные путем решения конкретной задачи динамики разреженного газа по методу прямого статистического моделирования, связанному с моделированием реализаций ансамбля тестовых частиц. На кластере НКС-30Т Сибирского суперкомпьютерного центра с использованием библиотеки PARMONC [8], был произведен ряд расчетов для общего числа ядер от 48 до 968. Реальные затраты машинного времени на независимое моделирование реализаций на ядрах-«вычислителях» и обмен данными (выборочными средними) с главным ядром-«сборщиком» были использованы для калибровки имитационной модели в AGNES.

Предполагаем, что архитектура предполагаемой суперЭВМ - однородный MPP-кластер. Масштабируемость определим следующим образом. Поскольку точность вычислений статистических методов зависит от количества независимых реализаций, поэтому увеличивая количество вычислителей (соответственно увеличивая количество реализаций в единицу времени), ожидаем пропорциональное уменьшение общего времени счета, при заданном уровне погрешности.

По результатам расчетов был сделан вывод, что требуемый уровень относительной статистической погрешности в 0.1% достигается при объеме выборки L, равном 240 000. Среднее время моделирования одной реализации составило 12 сек. Для ядер-«вычислителей» обмен данными с главным ядром-«сборщиком» происходил после каждой смоделированной на них реализации.

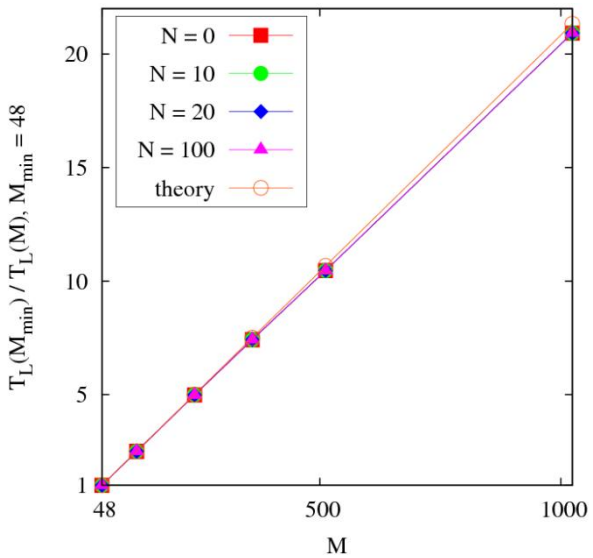

Рис. 2. Сравнение ускорения до M=1000. Результаты ускорения для модели совпадают с ускорением при расчётах с использованием PARMONC.

На рис. 2 приведены результаты реальных и модельных расчетов. Видно, что теоретическая оценка, практическая реализация и модельная при числе ядер-«сборщиков» от 10 до 100 практически совпадают.

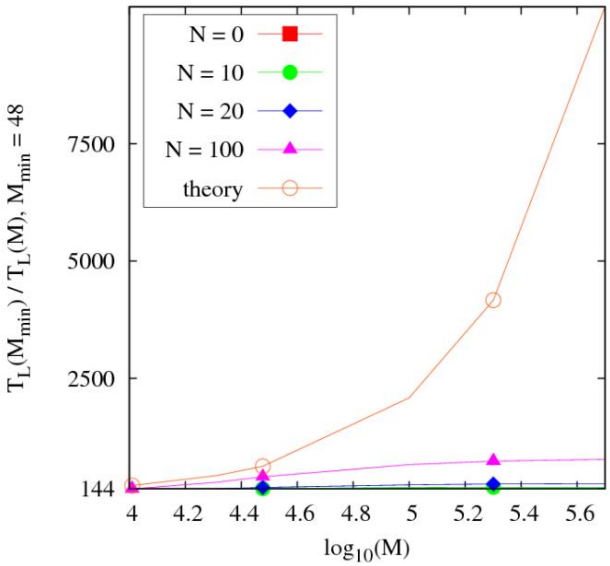

Рис. 3. Сравнение ускорения распределенного статистического моделирования для разных вариантов организации обмена данными для числа ядер M до 500 000 (горизонтальная ось – в логарифмическом масштабе).

Однако, как видно рис. 3, при значительном увеличении количества ядер ускорение резко падает. Промежуточные расчеты показывают, что уже при 2,5 тыс. ядер начинается расхождение с теоретической кривой, следовательно, нужно менять схему вычислений: либо увеличивать количество ядер-«сборщиков», либо строить из них дерево.

Аналогичные расчеты были проведены для другого класса алгоритмов, связанного с сеточными методами. Решалась задача численного моделирования распространения сейсмических полей в 3D изотропной неоднородной упругой среде [9]. В этом случае предполагаем, что архитектура гипотетического кластера является гибридной, вычислительные узлы состоят из нескольких CPU и GPU. Под масштабируемостью понимаем следующее - время счета алгоритма меняется незначительно при следующих допущениях: размер 3D модели увеличивается пропорционально количеству вычислительных узлов; каждый вычислительный узел совершает одно и то же количество итераций для своей подобласти.

Разработана программа на основе масштабируемого параллельного алгоритма при использовании комбинации технологий программирования CUDA и MPI. Для проведения расчетов различных 3D моделей была рассмотрена следующая организация параллельного алгоритма и программы: 3D область моделирования разделяется на слои, каждый слой рассчитывается независимо на выделенном GPU, а обмены данными между соседними GPU проводятся посредством MPI. При этом вычисления для слоя производятся посредством CUDA в 2D.

Для имитации сеточных методов реализован класс функциональных агентов Grid — узел-вычислитель, имитирующий расчет сеточных методов на одном вычислителе. Моделируются вычисления, когда область исследование режется вдоль одной оси, и полученные области загружаются на вычислители. Таким образом, получается, что у каждого вычислителя есть пересечение по данным максимум с 2-ми вычислителями («крайние» вычислители обмениваются только с одним соседом). Каждый вычислитель, на первом шаге рассчитывает свои граничные области, затем асинхронно передает насчитанные результаты соседям. Расчет внутренних областей

идет на втором шаге, получив данные от соседей и просчитав изменение свой области, агент переходит к шагу один.

Общие результаты изменения времени счета в зависимости от количества доступных ядер GPU (при пропорциональном увеличении размера 3D модели) в логарифмическом масштабе приведены на рис. 4. Показано хорошее соответствие экспериментальных и модельных результатов на начальном участке кривой (до 30720 ядер). При значительном увеличении количества вычислительных узлов с пропорциональным увеличением размера 3D модели время счета увеличивается, но несущественно (при росте числа узлов от 7680 до 1024000 время увеличилось на 17,5%).

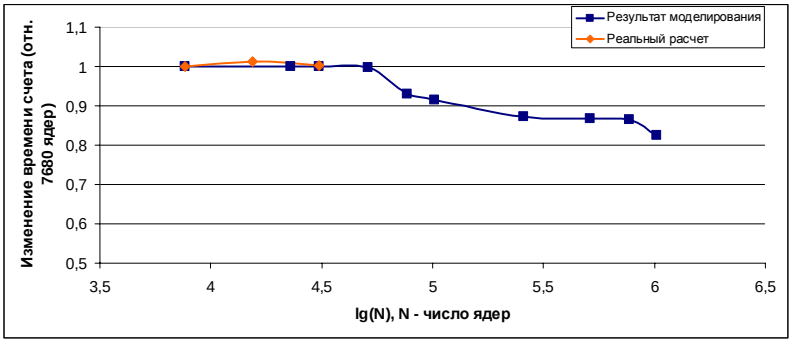

Рис. 4. Изменение времени расчета алгоритма численного моделирования в зависимости от числа вычислительных ядер (горизонтальная ось – в логарифмическом масштабе).

Проведенные численные эксперименты по имитационному моделированию показали возможность масштабирования алгоритмов на большое число (сотни тысяч и даже миллионы) вычислительных ядер предполагаемого экзафлопсного суперкомпьютера, а также возможность исследования поведения алгоритмов при таком большом масштабировании.

Дальнейшее направление исследований будет заключаться в разработке новых эффективных методов имитационного моделирования основных вычислительных алгоритмов из различных областей знаний, оценке их масштабируемости. Планируются исследования, на основе имитационного моделирования, производительности перспективных архитектур суперЭВМ при их наращивании до миллионов вычислительных ядер. По результатам моделирования будет дана сравнительная оценка перспективности расширения исследуемых архитектур с целью уменьшения издержек при разработке суперЭВМ экзафлопсной производительности и соответствующего программного обеспечения.

Литература

1. Peter Kogge(Ed.). ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems, DARPA report, September 28, 2008. Электронный адрес http://www.cse.nd.edu/Reports/2008/TR-2008-13.pdf

2. В.П. Иванников, С.С. Гайсарян, А.И. Аветисян, В.А. Падарян. Оценка динамических характеристик параллельной программы на модели // Программирование, №4, 2006., С. 21-37.

3. В.П. Иванников, А.И. Аветисян, С.С. Гайсарян, В.А. Падарян. Прогнозирование производительности MPI-программ на основе моделей // Журнал "Автоматика и телемеханика", 2007, №5, С. 8-17.

4. Б.М. Глинский, А.С. Родионов, М.А. Марченко, Д.И. Подкорытов, Д.В. Винс. Агентно-ориентированный подход к имитационному моделированию суперЭВМ экзафлопсной производительности в приложении к распределенному статистическому моделированию // Вестник ЮУрГУ, 2012. № 18 (277), Вып. 12., с. 93-106

5. Д.И. Подкорытов. Агентно-ориентированная среда моделирования сетевых систем AGNES // Ползуновский вестник, 2012. № 2/1, с. 93-106.

6. С.А. Степаненко, В.В. Южаков. Эксафлопные суперЭВМ: контуры архитектуры // Труды шестой международной конференции «Параллельные вычисления и задачи управления», 2012, с. 67-81. (ISBN 978-5-91450-122-5)

7. М.А. Марченко, Г.А. Михайлов Распределенные вычисления по методу Монте-Карло // Автоматика и телемеханика. 2007. Вып. 5. С. 157–170.

8. M.A. Marchenko PARMONC - A Software Library for Massively Parallel Stochastic Simulation // LNCS. 2011. V. 6873. P. 302-315.

9. Глинский Б.М., Караваев Д.А., Ковалевский В.В., Мартынов В.Н. Численное моделирование и экспериментальные исследования грязевого вулкана «Гора Карабетова» вибросейсмическими методами. //Вычислительные методы и программирование. М.: Изд-во Моск. Гос. ун-та, 2010, Том 11, №1, С. 99-108

ОБ АВТОРАХ

Глинский Борис Михайлович, зав. лаб. ССКЦ ИВМиМГ СО РАН. Новосибирский государственный университет, 1967. Д-р. тех. наук. 168 печатных работ, 4 монографии. Иссл. в обл. архитектур высокопроизводительных выч. систем и параллельных вычислений в геофизике. gbm@sscc.ru, 8(383)330-62-79

Glinskiy Boris, Head of department of Siberian Supercomputer Center ICMMG SB RAS. Novosibirsk State University, 1967. Phd, 168 articles, 4 monographs. Interested in architectures of high performance computers and parallel computing in geophysics. E-mail: gbm@sscc.ru, phone: +7 383 3306279

Родионов Алексей Сергеевич, зав. лаб. МоДПрИС ИВМиМГ СО РАН. Новосибирский электротехнический институт, 1976. Д-р. тех. наук. 84 печатные работы. Иссл. в обл. математического и имитационного моделирования систем сетевой структуры. alrod@sscc.ru, 8(383)330-69-49.

Rodionov A.S. Head of department of dynamic processes simulation in informational networks. Novosibirsk State Technical University, 1976. Phd, 84 articles. Interested in mathematical and simulation modeling of network structured systems. Email: alrod@sscc.ru, phone: +7 383 3306949.

Михайленко Борис Григорьевич, директор ИВМиМГ СО РАН. Новосибирский государственный университет, 1971 (год окончания ВУза).

Академик РАН, д-р физ.-мат. наук. 190 печатных работ, 3 монографии. Иссл. в обл. в области математического моделирования и создания новых численных методов решения задач геофизики. mikh@sscc.ru, 8 (383) 330-83-53

Mikhailenko Boris, Director of ICMMG SB RAS. Novosibirsk State University, 1971. Academician of RAS, Prof., 190 articles, 3 monographs. Interested in applied mathematics and computational geophysics. E-mail: mikh@sscc.ru, phone: +7 383 3308353

Марченко Михаил Александрович, ученый секретарь ИВМиМГ СО РАН. Новосибирский государственный университет, 1996 (год окончания ВУза). К-т физ.-мат. наук. 40 печатных работ. Иссл. в обл. параллельных алгоритмов статистического моделирования. marchenko@sscc.ru, 8 (383) 330-76-90

Marchenko Mikhail, academic secretary of ICMMG SB RAS. Novosibirsk State University, 1996. PhD, 40 articles. Interested in parallel algorithms in stochastic simulation. E-mail: marchenko@sscc.ru, phone: +7 383 3307690

Черных Игорь Геннадьевич, н.с. ИВМиМГ СО РАН. Новосибирский государственный университет, 2002. к.ф-м.н. 30 печатных работ. Исследования в области архитектур высокопроизводительных архитектур, параллельные вычисления в области химической кинетики и фотокатализе. chernykh@ssd.sscc.ru, 8(383)330-96-65

Chernykh Igor, senior researcher ICMMG SB RAS. Novosibirsk State University, 2002. Phd, 30 articles. Interested in architectures of high performance computers and parallel computing in chemical kinetics and photocatalysis. E-mail: chernykh@ssd.sscc.ru, phone: +7 383 3309665.