

**Всероссийская летняя XXXVII
молодежная Школа-конференция по
параллельному программированию
с международным участием**

(г. Новосибирск, 5 – 16 июля 2021 г.)

**Всероссийская летняя XXXIX
молодежная Школа-конференция по
параллельному программированию
с международным участием**

(г. Новосибирск, 4 – 15 июля 2022 г.)

Тезисы докладов

ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ
И МАТЕМАТИЧЕСКОЙ ГЕОФИЗИКИ СО РАН
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Всероссийская летняя XXXVII молодежная
Школа-конференция по параллельному программированию
с международным участием
(г. Новосибирск, 5 – 16 июля 2021 г.)

Всероссийская летняя XXXIX молодежная
Школа-конференция по параллельному программированию
с международным участием
(г. Новосибирск, 4 – 15 июля 2022 г.)

Тезисы докладов

Новосибирск
2023

УДК 004.4
ББК 32.973
В85

В85 Всероссийская летняя XXXVII молодежная Школа-конференция по параллельному программированию с международным участием (г. Новосибирск, 5 – 16.07.2021 г.). Всероссийская летняя XXXIX молодежная Школа-конференция по параллельному программированию с международным участием (г. Новосибирск, 4 – 15.07.2022 г.): Тез. докл. / отв. ред. д-р техн. наук, проф. В. Э. Малышкин. Новосиб. гос. унт. – Новосибирск : ИПЦ НГУ, 2023. – 44 с.

ISBN 978-5-4437-1460-8

В сборнике представлены тезисы докладов участников двух Всероссийских летних молодежных Школ-конференций по параллельному программированию с международным участием, которые были организованы ИВМиМГ СО РАН, НГУ и НГТУ и проходили в Новосибирском государственном университете с 5 по 16 июля 2021 г. и с 4 по 15 июля 2022 г.

Для студентов, магистрантов, аспирантов и молодых ученых, использующих высокопроизводительные вычислительные ресурсы в научной и учебной деятельности.

УДК 004.4
ББК 32.973

ISBN 978-5-4437-1460-8
DOI 10.25205/978-5-4437-1460-8

© Институт вычислительной математики
и математической геофизики СО РАН, 2023
© Новосибирский государственный
университет, 2023
© Новосибирский государственный
технический университет, 2023

ПРЕДИСЛОВИЕ

Организаторами Всероссийской XXXVII летней молодежной Школы-конференции по параллельному программированию с международным участием (URL: <https://ssd.sccc.ru/ru/school/2021s>) и Всероссийской XXXIX летней молодежной Школы-конференции по параллельному программированию с международным участием (URL: <https://ssd.sccc.ru/ru/school/2022s>) являлись Институт вычислительной математики и математической геофизики СО РАН (ИВМиМГ СО РАН), Новосибирский государственный университет (НГУ) и Новосибирский государственный технический университет (НГТУ). Школы-конференции были проведены при поддержке Министерства образования и науки РФ, Сибирского отделения Российской академии наук, Совета по суперкомпьютерным вычислениям СО РАН.

Основная цель школы-конференции – дать возможность студентам, магистрантам, аспирантам и молодым ученым продемонстрировать собственные результаты и разработки, обсудить актуальные фундаментальные научные проблемы в области создания технологий параллельного программирования и их применения для решения практических задач, сформировать и укрепить научные контакты. В рамках программ школы-конференции выступили с докладами известные ученые, представив обзор современной проблематики в сфере технологий параллельного программирования и тенденций их развития.

Школа-конференция является частью систематической работы организаторов по развитию научной коммуникации и научных кадров в области проблем параллельного программирования. В сотрудничестве с партнерами раз в два года, начиная с 1991 г., проводится Международная научная конференция «Parallel Computing Technologies» (PaCT, <http://ssd.sccc.ru/en/conference>). Ежегодно с 2000 г. ИВМиМГ СО РАН, НГУ и НГТУ организуют зимнюю и летнюю школы по параллельному программированию для приходящих в науку студентов, с 2017 года летняя школа трансформировалась в школу-конференцию, программа которой составляется из лекций ведущих ученых, двухнедельных образовательных проектов и конференционной части с докладами участников о результатах предложенной организаторами проектной работы и собственных научных исследований в период до конференции.

Школа-конференция специализируется в области научных проблем, связанных с развитием технологий решения задач на параллельных вычислительных системах. Приложения параллельного программирования в таком контексте рассматриваются постольку, поскольку позволяют сформулировать требования к развитию технологий программирования,

организации управления данными и вычислениями и оценить технологии с точки зрения применимости к реальным задачам.

Высокопроизводительные вычислительные системы дают возможность проводить крупномасштабное численное моделирование и обработку большого объема данных, обеспечивая развитие науки, решение практических задач промышленности и государственного управления. Вместе с тем с учетом современного состояния технологий организации вычислений на суперкомпьютерах разработчику приложений нужно владеть теорией и практикой системного параллельного программирования, быть способным разрабатывать корректные параллельные алгоритмы и применять низкоуровневые средства программирования, интерфейсы управления данными и вычислительными задачами. По этой причине затруднено эффективное применение высокопроизводительных вычислительных систем прикладными специалистами, сдерживается развитие науки и технологий.

Школы-конференции ИВМиМГ СО РАН, НГУ, НГТУ по параллельному программированию способствуют продвижению исследований в области актуальной проблемы создания высокоуровневых инструментов параллельного программирования, в частности предметно-ориентированных систем программирования, систем для автоматизированного накопления и переиспользования алгоритмов и программ, параллельных алгоритмов для решения конкретных прикладных задач, повышения эффективности работы вычислительных центров.

**ПРОГРАММА ЛЕКЦИЙ
ВСЕРОССИЙСКОЙ ЛЕТНЕЙ XXXVII МОЛОДЕЖНОЙ ШКОЛЫ-
КОНФЕРЕНЦИИ ПО ПАРАЛЛЕЛЬНОМУ
ПРОГРАММИРОВАНИЮ С МЕЖДУНАРОДНЫМ УЧАСТИЕМ**

Будущие технологии параллельного программирования

Малышкин Виктор Эммануилович,
д-р техн. наук, гл. науч. сотр., зав. лаб. ИВМиМГ СО РАН

В настоящее время в программировании осуществляется переход от пассивных знаний к активным. Будущие технологии параллельного программирования будут основываться на базах активных знаний. Главное отличие такой базы от библиотеки подпрограмм состоит в том, что модулем базы знаний является подпрограмма, формально включённая в описание некоторой предметной области. Благодаря этому подпрограмма может выбираться и исполняться автоматически, если это требуется для проведения вычислений, которые запросил пользователь базы активных знаний. Создание теоретических основ и прототипирование баз активных знаний является актуальной задачей современного системного параллельного программирования.

Технология фрагментированного программирования и система LuNA

Перепёлкин Владислав Александрович, науч. сотр. ИВМиМГ СО РАН

В ИВМиМГ СО РАН разрабатывается язык и система LuNA автоматического конструирования параллельных программ. Проект направлен на приложение концепции активных знаний к автоматизации параллельного программирования. Команда разработчиков включает как сотрудников института, так и студентов и аспирантов. В проекте решается целый спектр передовых научных проблем в области системного параллельного программирования и автоматического синтеза программ. Лекция посвящена введению в проблематику создания системы LuNA, текущим достижениям и дальнейшим планам.

**Прямые методы решения разреженных СЛАУ:
особенности параллелизации**

Соловьев Сергей Александрович,
канд. физ.-мат. наук, ст. науч. сотр. ИНГГ СО РАН

В лекции рассказывается, как в различных прикладных областях возникают задачи, связанные с решением систем линейных

алгебраических уравнений. Кратко показано преимущество прямых методов над итерационными. Более подробно рассказывается о реализации прямых методов на параллельных вычислительных системах. Показано, как дополнительный подготовительный этап может существенно уменьшить потребляемую память, число арифметических операций, а также повлиять на возможность распараллеливания. Обсуждаются подходы к реализации алгоритмов на системах с распределенной и общей памятью. В заключение предлагается обсудить проблемы и пути развития обсуждаемых подходов.

Высокопроизводительные вычисления на базе Desktop Grid

Ивашко Евгений Евгеньевич, канд. физ.-мат. наук, ст. науч. сотр.
Института прикладных математических исследований КарНЦ РАН,
руководитель ЦКП КарНЦ РАН «Центр высокопроизводительной
обработки данных», г. Петрозаводск

Высокопроизводительные вычисления играют все возрастающую роль при проведении научных исследований, разработке новых видов промышленной продукции и в социальной сфере. Для выполнения высокопроизводительных расчетов, как правило, используются вычислительные кластеры или – когда требуются особенно большие объемы ресурсов – грид-системы (яркий пример – LHC Computing Grid), объединяющие вычислительные кластеры. Однако с развитием каналов связи (и соответствующим увеличением скорости и пропускной способности) сети Интернет и ростом производительности персональных компьютеров становится все более популярным направление, связанное с организацией Desktop Grid – грид-сетей, объединяющих неспециализированные вычислители (как правило, персональные компьютеры). Desktop Grid позволяют достаточно быстро и легко объединить в «виртуальный суперкомпьютер» значительное число источников сравнительно небольших вычислительных ресурсов для решения вычислительноемких задач. Desktop Grid служит дополнением к вычислительным кластерам и классическим грид-системам.

В докладе будет рассказано о месте Desktop Grid систем в области высокопроизводительных вычислений, особенностях и ограничениях таких систем, прикладных задач, решаемых с их помощью, а также об основных современных задачах, связанных с развитием Desktop Grid.

Проблемы отладки и оптимизации параллельных программ. Методы решения

Власенко Андрей Юрьевич,
канд. техн. наук, мл. науч. сотр. ИВМиМГ СО РАН

В лекции рассматриваются фундаментальные для параллельных программ проблемы отладки и оптимизации. Несмотря на то что параллельные вычисления развиваются уже не одно десятилетие, эти проблемы еще пока очень далеки от качественных универсальных решений. Приводится классификация семантических ошибок для наиболее распространенной технологии программирования систем с распределенной памятью – MPI. Излагаются методы автоматизированной отладки в сопоставлении с классической диалоговой отладкой и друг с другом. Кратко охарактеризованы популярные системы отладки. Даются возможные цели оптимизации параллельных программ, приемы оптимизации и инструментальные средства, помогающие обнаружить и исключить неоптимальное поведение программы. Разбираются примеры “плохих” программ, нуждающихся в оптимизации.

Применение нейронных сетей при решении геофизических задач

Дучков Антон Альбертович, канд. физ.-мат. наук, зав. лабораторией динамических проблем сейсмологии ИНГГ СО РАН, доц. каф. геофизики ГГФ НГУ

Представлен опыт применения нейронных сетей к решению задач сегментации томографических изображений, определения границ слоев по данным геофизических исследований в скважинах, определения первых вступлений волн, выделения разломов.

Решение уравнения эйконала с помощью нейронной сети для моделирования времен пробега сейсмических волн. Исследование параметров архитектуры, влияющих на устойчивость получаемого решения

Грубась Серафим Игоревич, инженер ИНГГ СО РАН

Представлены результаты исследования применения Physics-Informed Neural Networks (PINN) к обработке микросейсмических данных в задаче локализации микросейсмических событий. Предлагаемый алгоритм включает решение уравнения эйконала с помощью обученной на результатах численного моделирования нейронной сети.

Численный прогноз опасных метеорологических условий с использованием суперкомпьютера

Старченко А. В., Проханов С. А., Данилкин Е. А., Лещинский Д. В.
Докладчик: Старченко Александр Васильевич, д-р физ.-мат. наук, проф.,
зав. каф. вычислительной математики и компьютерного моделирования
ТГУ, вед. науч. сотр. регионального научно-образовательного
математического центра ТГУ

Представлена мезомасштабная метеорологическая модель высокого разрешения для прогноза и исследования погодных явлений и качества приземного воздуха над ограниченной урбанизированной территорией, крупным промышленным или транспортным узлом. Для решения уравнений мезомасштабной модели разработан экономичный явно- неявный разностный метод второго порядка аппроксимации, ориентированный на многопроцессорную вычислительную систему. Результаты тестирования параллельного кода на кластере ТГУ Cyberia показали его высокую эффективность. Приведены примеры успешного применения по численному предсказанию интенсивных осадков, сильного ветра и тумана.

Interpretable ML для анализа мультимодальных данных психометрии и нейрофизиологии

Рудыч Павел Дмитриевич,
науч. сотр. НИИ нейронаук и медицины СО РАН

Рассматриваются характерные классы реальных задач, лежащих на стыке ML и психофизиологии. В качестве демонстрационных характерных примеров рассмотрены психометрические данные опросников и мультимодальные данные диагностики депрессий. Данные объединены в таблицу и проанализированы с помощью инструментов interpretable ML.

**ПРОГРАММА ЛЕКЦИЙ
ВСЕРОССИЙСКОЙ ЛЕТНЕЙ XXXIX МОЛОДЕЖНОЙ ШКОЛЫ-
КОНФЕРЕНЦИИ ПО ПАРАЛЛЕЛЬНОМУ
ПРОГРАММИРОВАНИЮ С МЕЖДУНАРОДНЫМ УЧАСТИЕМ**

**Система автоматического конструирования
параллельных программ LuNA**

Перепёлкин Владислав Александрович, науч. сотр. ИВМиМГ СО РАН

Автоматическое конструирование параллельных программ облегчает разработку и отладку в численном моделировании, а в потенциале позволяет и повысить эффективность параллельных программ по сравнению со средней ручной разработкой. Также оно позволяет автоматически обеспечивать динамическую балансировку нагрузки, отказоустойчивость и другие нефункциональные свойства. Автоматизация конструирования параллельных программ требует решения комплекса различных задач. Многие из них, а также возможности их решения видны на примере системы LuNA, которой посвящена лекция.

**Новое оборудование ССКЦ СО РАН для обучения нейросетевых
моделей**

Городничев Максим Александрович, науч. сотр. ИВМиМГ СО РАН

В лекции представлен обзор нового для ССКЦ СО РАН оборудования: серверов Хуавей, построенных на процессорах Kunpeng архитектуры ARM, а также платах расширения с нейропроцессорами Ascend архитектуры Da Vinci. Представлен опыт реализации нейросетевых моделей на этом оборудовании с использованием фреймворка MindSpore.

**Дискретные методы в задачах моделирования
пространственной динамики природных процессов**

Медведев Юрий Геннадьевич,
канд. техн. наук, ст. науч. сотр. ИВМиМГ СО РАН

В лекции рассказывается о дискретном подходе в области имитационного моделирования на примере клеточных автоматов. Рассмотрены модели двумерной и трехмерной диффузии / теплопроводности, ламинарных и турбулентных потоков жидкостей и газов, просачивания жидкости через пористую среду, распространения ударной волны, затронуты вопросы динамических граничных условий. Описаны способы последовательной и параллельной программной реализации таких моделей, рассказано о методах балансировки нагрузки вычислительного кластера.

ТЕЗИСЫ ДОКЛАДОВ УЧАСТНИКОВ

Проектирование подсистемы профилирования для системы фрагментированного программирования LuNA

Е. С. Абрамушкина

Новосибирский государственный университет
e.abramushkina@g.nsu.ru

Одним из направлений работ в ИВМиМГ СО РАН является создание методов и средств параллельной реализации больших численных моделей на суперкомпьютерах. В связи с этим в ИВМиМГ СО РАН разрабатывается технология фрагментированного программирования, в частности, реализующие её язык и система программирования LuNA [1, 2], предназначенные для поддержки разработки параллельных программ, реализующих большие численные модели.

В рамках проекта по разработке системы LuNA есть множество подпроектов, связанных со сбором профиля исполняемой LuNA-программы, то есть сбором информации о событиях, происходящих в процессе исполнения программы. В настоящее время каждый подпроект использует собственную реализацию подсистемы профилирования, разработанную с учетом того, какая информация о выполнении программы необходима для дальнейшего анализа. Информация, содержащаяся в профиле, используется для автоматической балансировки нагрузки в системе [3], для оценки эффективности фрагментации алгоритма [4], для обнаружения причин зависания программ [5], для динамической балансировки вычислительной нагрузки при воспроизведении трасс [6]. Поскольку для различных целей требуется собирать профиль исполняемых параллельных программ, было принято решение унифицировать формат профиля и разработать в системе фрагментированного программирования LuNA единую подсистему профилирования.

В ходе обзора работ, связанных с профилированием LuNA-программ, участниками проекта по разработке системы LuNA был выдвинут ряд требований, которым должна удовлетворять подсистема профилирования. В результате анализа данного списка можно заключить, что требования к подсистеме профилирования можно разделить на три класса.

1. Помимо возможности записи информации о событиях в системе требуется обеспечить возможность промежуточной обработки данной информации, её накопления, использования другими подсистемами и вывода в различных форматах в различные места (экран, файл, сеть).
2. Необходимо обеспечить расширяемость функциональности подсистемы и используемых форматов представления информации.

3. Накладные расходы [7, с. 18] на работу подсистемы профилирования нужно минимизировать.

На основе требований, которым должна удовлетворять подсистема, были приняты следующие проектные решения.

1. Спроектировать подсистему профилирования системы LuNA в виде расширяемого набора модулей.
2. Спроектировать базовый формат представления профилировочной информации, который в дальнейшем будет расширяться.
3. Обеспечить минимизацию накладных расходов при помощи параметризации компиляции и запуска системы LuNA. Параметры будут определять требуемую функциональность подсистемы.

К настоящему времени предложена модель представления профилировочной информации и её текстовое представление на основе формата JSON. Реализован первый прототип подсистемы профилирования LuNA-программ. Прототип имеет модульную структуру и частично удовлетворяет поставленным требованиям. В дальнейшем он будет доработан в соответствии с описанными решениями.

Литература

1. Голиков М. О. Разработка и реализация средств для визуального анализа исполнения фрагментированных программ: вып. квалиф. работа. Новосибирск: НГУ, 2021.
2. Литвинов В. С. Разработка алгоритмов анализа производительности фрагментированных программ: вып. квалиф. работа. Новосибирск: НГУ, 2017.
3. Лямин А. С. Разработка и реализация алгоритма распределения ресурсов фрагментированных программ на основе профилирования: вып. квалиф. работа. Новосибирск: НГУ, 2021.
4. Малышкин В. Э. Технология фрагментированного программирования // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. Челябинск: ЮУрГУ, 2012. № 46 (305). С. 45–55.
5. Мичуров М. А. Средство анализа причин зависаний фрагментированных программ в системе LuNA // Инновации. Наука. Образование. Гольягти, 2021. № 40. С. 354–364.
6. Саяпин М. П. Разработка и реализация алгоритмов динамической балансировки вычислительной нагрузки для подсистемы воспроизведения трасс в системе LuNA: вып. квалиф. работа. Новосибирск: НГТУ, 2022.
7. Kireev S., Malyshkin V., Fujita H. The LuNA Library of Parallel Numerical Fragmented Subroutines // Proceedings of the 11th International Conference on Parallel Computing Technologies, LNCS. 2011. Vol. 6873. P. 290–301.

Научный руководитель – С. Е. Киреев, ИВМиМГ СО РАН, НГУ

Случайный сферический процесс дрефта-диффузии. Сравнение OpenMP- и MPI-подходов

И. А. Аксюк

Новосибирский государственный университет
Институт вычислительной математики и математической геофизики
СО РАН, г. Новосибирск
i.aksyuk@g.nsu.ru

Процесс дрефта-диффузии встречается во многих практических задачах, например, при расчете захвата малых частиц массивным телом, расчете емкости в электростатике, гидродинамического трения и скорости реакции макромолекул и во многих других примерах.

В данной работе рассматривался процесс блуждания экситонов в бесконечном слое полупроводника с известными параметрами диффузии, скорости движения и среднего времени жизни экситонов. Уравнение дрефта-диффузии описывает концентрацию экситонов в слое. Для решения уравнения дрефта-диффузии мы использовали метод случайного блуждания по сферам в обобщенном виде, предложенном в работе [1]. Траектории движения частиц здесь моделируются шаг за шагом в последовательности специально построенных случайных сфер. При этом случайное распределение на поверхности сферы моделируется согласно распределению вероятностей фон Мизеса–Фишера. Процесс последовательного моделирования множества траекторий частиц занимает длительное время, но этот процесс можно эффективно распараллелить.

В данной работе были продемонстрированы результаты распараллеливания процесса блуждания частиц при решении уравнения дрефта-диффузии с помощью OpenMP- и MPI-подходов. Тесты проводились на кластере Информационно-вычислительного центра НГУ на 12-ядерном и 24-ядерном узлах. Распараллеливание с помощью OpenMP дало ускорение в 18,46 раз с эффективностью 76,94 % на 24-ядерном узле и в 11,17 раз с эффективностью 93,14 % на 12-ядерном узле. Распараллеливание с помощью MPI дало ускорение в 20,87 раз с эффективностью 86,98 % на 24-ядерном узле и в 11,44 раз с эффективностью 95,35 % на 12-ядерном узле.

Работа выполнена в рамках гранта РФФИ 19-11-00019.

Литература

1. Sabelfeld K. K. Random walk on spheres method for solving drift-diffusion problems // Monte Carlo Methods and Applications. 2016. Vol. 22. P. 265–281.

Научный руководитель – д-р физ.-мат. наук, проф. К. К. Сабельфельд,
ИВМиМГ СО РАН, НГУ

Разработка экспериментальной библиотеки работы с распределенным массивом для поддержки автоматического конструирования параллельных программ

И. Н. Баранов

Новосибирский государственный университет
i.baranov@g.nsu.ru

В настоящее время написание параллельных программ является сложной задачей. В основном возникают сложности с организацией взаимодействия между процессами: дедлоки, синхронизация, состояние гонки данных и другие. Одним из способов кардинально решить эти проблемы является синтез параллельных программ.

В задаче синтеза есть проблема подбора параметров, которые влияют не на результат работы программы, а на её нефункциональные характеристики, такие как время работы, потребление памяти и другие. Такие параметры будем называть параметрами реализации. Подобрать приемлемые значения параметров – не всегда простая задача в связи с большим количеством различных комбинаций этих параметров. Эта задача может быть упрощена в случае, когда есть знания о влиянии параметров на эффективность работы программы.

Чтобы проработать проблему подбора параметров, ограничимся простым классом задач с простыми параметрами, влияние которых относительно известно: задачи пространственной динамики на регулярных сетках. Для их параллельной реализации будем использовать распределенный массив.

Целью работы является разработка параллельной библиотеки работы с распределенным массивом, а также средства для автоматизации процесса подбора ее параметров на основе знаний эксперта.

Был проведен анализ существующих библиотек, решающих поставленную задачу, и на его основе спроектировано и реализовано собственное решение. Разработанная библиотека поддерживает распределенный массив с разными вариантами его декомпозиции. Есть возможность менять некоторые параметры реализации библиотеки, такие как количество процессов, потоков, тип декомпозиции и другие.

В данной работе был выбран подход с использованием собственного скриптового языка для представления знаний эксперта. Скриптовый язык позволяет описывать множество параметров программы (задавать диапазоны значений, перечисления, их комбинации), указывать метод подбора значений параметров (перебор по сетке, случайный поиск, с помощью другого скрипта), ограничивать пространство перебора с помощью фильтрации (логические выражения, удаление комбинаций по

строке, оставление N комбинаций), задавать зависимости между параметрами в виде формулы. Написанный скрипт поступает на вход разработанному средству подбора параметров, а на выходе получаем оптимальную комбинацию значений параметров. Средство подбора параметров поддерживает запуск расчетов на локальной системе или на кластере, имеет модуль визуализации результатов.

Разработанное средство было протестировано на программе решения двумерного уравнения Гельмгольца методом Якоби, написанной с помощью библиотеки.

В результате работы были разработаны и протестированы библиотека параллельного программирования для работы с распределенными массивами и средство подбора параметров, которые можно внедрять в системы автоматического конструирования параллельных программ.

Научный руководитель – С. Е. Киреев, ИВМиМГ СО РАН, НГУ

Распараллеливание с помощью OpenMP и MPI глобального стохастического алгоритма блуждания по сетке для решения нелинейной системы уравнений Бюргерса

О. В. Бухашеев

Новосибирский государственный университет
Институт вычислительной математики и математической геофизики
СО РАН, г. Новосибирск

Система уравнений Бюргерса представляет собой особую форму системы уравнений Навье-Стокса для несжимаемой жидкости без давления и уравнения неразрывности. Уравнения Бюргерса являются важными дифференциальными уравнениями в частных производных, широко используемыми для различных физических приложений, таких как моделирование гидро- и газодинамики, транспортных потоков, ударных волн.

В формуле ниже G – ограниченная область с границей ∂G . Численные эксперименты проводятся на примере единичного квадрата. В данной работе для решения этой начально-краевой задачи применяется глобальный алгоритм случайного блуждания по сетке [1-3], который в отличие от других алгоритмов блуждания вычисляет решение в любом желаемом семействе n заданных точек. Система из уравнений Бюргерса является нелинейной, поэтому классические стохастические методы блуждания по сферам неприменимы к ней. Идея основана на итерационном по времени процессе нахождения решения системы с

известным полем скорости, которое вычисляется из решения на предыдущем шаге.

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial t} = D\Delta u - u \frac{\partial u}{\partial x} - v \frac{\partial u}{\partial y}, \quad \bar{x} \in G \times [0, t_{\max}] \\ \frac{\partial v}{\partial t} = D\Delta v - u \frac{\partial v}{\partial x} - v \frac{\partial v}{\partial y}, \quad \bar{x} \in G \times [0, t_{\max}] \\ u(x, y, 0) = f(x, y) \\ v(x, y, 0) = g(x, y) \\ u|_{\partial G} = \varphi_1 \\ v|_{\partial G} = \varphi_2 \end{array} \right.$$

Численные решения сравниваются с точными решениями системы [4–6]. Анализируются точность и компьютерное время алгоритма.

Для распараллеливания последовательной программы были использованы библиотеки OpenMP и MPI. Для этого блуждающие частицы, на траекториях которых вычисляются монтекарловские оценки, были распределены по разным потокам, MPI–процессам соответственно. Тесты проводились на кластере Информационно-вычислительного центра НГУ на 24-ядерном узле. Подход с использованием OpenMP дал ускорение в 12 раз с эффективностью 35 % на 48 логических потоках. Распараллеливание с MPI дало ускорение в 14 раз с эффективностью 45 % на 48 логических потоках.

Работа выполнена в рамках гранта РФФ 19-11-00019.

Литература

1. Abazari R., Borhanifar A. Numerical study of the solution of the Burgers and coupled Burgers equations by a differential transformation method // *Computers & Mathematics with Applications*. 2010. Vol. 59. Issue 8, P. 2711–2722.
2. Hongqing Zhua, Huazhong Shub, Meiyu Ding. Numerical solutions of two-dimensional Burgers' equations by discrete Adomian decomposition method // *Computers and Mathematics with Applications*. 2010. Vol. 60. P. 840–848.
3. Kireeva A., Sabelfeld K. K., Kireev S. Parallel simulation of drift-diffusion-recombination by cellular automata and global random walk algorithm // *The Journal of Supercomputing*. 2021. Vol. 77. P. 6889–6903.

4. Sabelfeld K. K., Kireev S., Kireeva A. Parallel implementation of cellular automata model of electron-hole transport in a semiconductor // Journal of Parallel and Distributed Computing. 2021. Vol. 158, P. 186–195.
5. Sabelfeld K., Smirnov D., Dimov I., Todorov V. A global random walk on grid algorithm for second order elliptic equations // Monte Carlo Methods and Applications. 2021. Vol. 27. № 4. P. 325–339.
6. Bhardwaj U. Numerical Solutions to Burgers' Equations // B. Tech. thesis. CFAI University, Dehradun. 2011.

Научный руководитель – д-р физ.-мат. наук, проф. К. К. Сабельфельд

Управление распределенными фрагментированными данными в системе LuNA

А. А. Валитов

Новосибирский государственный университет

a.valitov@g.nsu.ru

В распределенных вычислительных системах обеспечение доступности данных, приемлемой скорости доступа к ним являются сложными задачами. В системах автоматического конструирования параллельных программ [1] актуальна проблема повышения эффективности исполнения генерируемых программ.

В общем случае для задачи управления распределенными данными не существует универсального алгоритма, генерирующего эффективную фрагментированную программу [2]. Многие алгоритмы управления распределенными данными обеспечивают удовлетворительный средний результат в своей области применения, однако они теряют эффективность на других классах задач. В частности, эффективность зависит и от степени доступности распределенных данных. Одним из решений проблемы может быть накопление и применение частных алгоритмов, которые на конкретных классах задач способны генерировать эффективную фрагментированную программу. Именно такого типа алгоритмы повышения доступности данных в системе LuNA могут повысить эффективность вычислений.

В системе LuNA задача повышения доступности данных заключается в снижении количества и времени миграций фрагментов данных между узлами.

На каждом из узлов производится исполнение фрагментов вычислений, которые представляют собой часть программы и могут быть исполнены при наличии соответствующих им входных данных. Этими данными в системе LuNA являются фрагменты данных, которые представляют собой переменные, структуры и массивы. В результате выполнения фрагмента

вычислений могут быть порождены новые фрагменты вычислений и данных. Фрагменты могут мигрировать между вычислительными узлами распределенной системы, уже находиться на них, отсутствовать или быть запрошены.

Для осуществления миграции фрагментов данных в системе существует два механизма: требующиеся данные запрашиваются одним узлом у другого и отправляются после обработки запроса, или же узел отправляет данные нуждающемуся узлу по готовности. Оба механизма используют т. н. локатор, то есть объект, отвечающий за формирование множества узлов, на которые необходимо отправить фрагмент данных.

Такие подходы показывают невысокую эффективность на практически значимых классах задач, в которых некоторые фрагменты данных требуются на многих (или даже на всех) узлах. Например, если задача подразумевает использование глобальных констант на всех узлах.

Для повышения эффективности вычислений на таких классах задач предлагается использовать алгоритмы частичной и массовой репликации фрагментов данных, то есть алгоритмы, позволяющие выполнять пересылку фрагмента данных как на все узлы, так и на определенное множество узлов системы. Массовая репликация закрывает проблему поиска данных и обеспечения приемлемой скорости их получения, поскольку узлам теперь не придется отправлять запрос на получение отсутствующего фрагмента данных. Частичная репликация находит свое применение в случае более строгих ограничений на память узлов системы или достаточной высокой скорости коммуникаций между ними. При таком подходе данные располагаются на определенном множестве узлов, что позволяет сэкономить память других узлов, а также уменьшить количество коммуникаций между ними в системе в целом.

Предложенные алгоритмы были реализованы и интегрированы в исполнительную систему LuNA, причем с применением частично уже реализованных методов системы. Это позволило расширить систему, почти никак не изменяя уже протестированную часть. Механизм репликации был реализован с помощью нового типа локатора. Новый локатор отвечает как за массовую, так и за частичную репликацию, изменяя лишь множество узлов и реализуя один интерфейс ответа исполнительной системе.

Для проведения тестирования эффективности предложенных алгоритмов была реализована программа, в которой большое количество фрагментов вычислений используют один и тот же фрагмент данных, изначально находящийся на одном узле. Результаты запуска тестовой фрагментированной программы на четырех процессорах показали уменьшение времени исполнения примерно на 13 % в случае массовой репликации и 8 % в случае частичной репликации.

Таким образом, предложенные частные алгоритмы управления распределенными данными позволили расширить класс задач, решаемых системой LuNA более эффективно по сравнению с предыдущими реализованными алгоритмами управления данными в системе. Предложенные частные алгоритмы снижают количество требуемых запросов между узлами и нагрузку на коммуникационную сеть, что в конечном счете позволяет повысить эффективность LuNA–программ.

В дальнейшем предполагается работа по получению характеристик исполнения прикладных задач, использующих предложенные алгоритмы репликации, и доработка их внутренней реализации.

Литература

1. Вальковский В. А., Малышкин В. Э. Синтез параллельных программ и систем на вычислительных моделях. Новосибирск: Наука. Сиб. отд-ние, 1988. 129 с.
2. Malyshkin V., Perepelkin V. Optimization methods of parallel execution of numerical programs in the LuNA fragmented programming system // The Journal of Supercomputing. 2012. Vol. 61. № 1. P. 235–248.

Научный руководитель – В. А. Перепёлкин, ИВМиМГ СО РАН, НГУ

Использование оперативной памяти для кеширования в системах с распределенной виртуальной памятью

С. В. Вязигин

Казахский национальный университет
им. Аль-Фараби, г. Алматы, Казахстан
wismas1996@gmail.com

Одной из перспективных задач совершенствования и развития вычислительных систем является задача оптимизации мультимьюльтикомпьютерных систем с распределенной виртуальной памятью [1]. При этом специалисты сталкиваются с ситуацией, когда мультимьюльтикомпьютерная система, включающая значительное количество вычислительных узлов, в определенных ситуациях теряет производительность из-за избыточного количества страничных отказов. Страничные отказы в основном происходят в момент, когда вычислительному узлу требуется страница или блок данных, находящийся или использующийся в текущий момент на другом вычислительном узле. На сегодняшний день активно разрабатываются новые средства [2] для повышения производительности подобных систем, и одним из возможных

решений по уменьшению количества страничных отказов и, как следствие, повышения производительности системы является кеширование страниц в оперативную память узла [3].

Для определения потенциала минимизации количества страничных отказов за счет кеширования страниц в оперативную память был проведен эксперимент, в котором замерялись как общее время выполнения программы, так и время каждой операции. Система с распределенной виртуальной памятью была построена основе FPGA в количестве 10 штук, соединенных друг с другом по сети LAN по протоколу TCP/IP. На рис. 1, *а* изображен график задержек всей системы, состоящей из 10 узлов FPGA, за период в 32 секунды, а на рис. 1, *б* изображен график задержек только одного из вычислительных узлов, выделенный из общего графика на рис. 1, *а*, в период с 3 до 4 секунд.

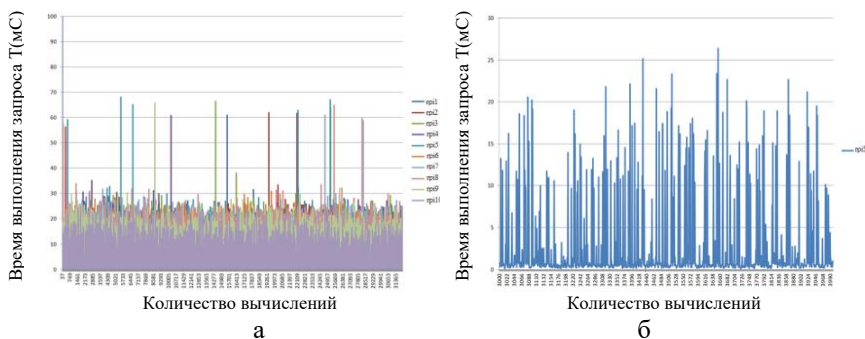


Рис. 1. График задержек без использования кеша: *а* – график задержек всей системы; *б* – график задержек n01

Как видно на рис. 1, среднее время отклика в данной системе составило 15 миллисекунд, а максимальное время достигало 68 миллисекунд. Для повышения производительности системы у каждого из узлов было выделено 5 % от оперативной памяти под кеш. Данный кеш позволяет дублировать страницы для дальнейшего чтения их из локальной памяти, а не обмена страниц в системе. Страницы в данном кеше помечены как «только для чтения», и в случае когда необходимо записать в эту страницу данные, текущий узел отправляет запрос на удаление страницы из других узлов, после чего данная страница остается в единственном экземпляре в системе и выполняются все требования для редактирования страниц в системе с распределенной виртуальной памятью [4]. На рис. 2 изображены графики задержек аналогично рис. 1 после добавления кеша в оперативную память.

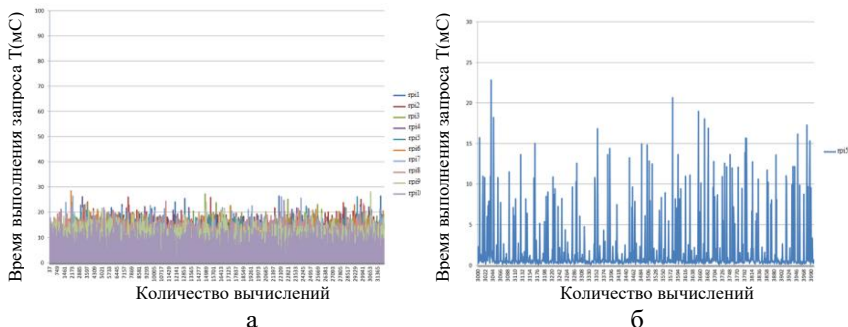


Рис. 2. График задержек с использованием кеша в оперативной памяти: *а* – график задержек всей системы; *б* – график задержек gr15

Как видно на рис. 2, после введения кеша в оперативной памяти средняя длительность страничных отказов уменьшилась с 15 миллисекунд до 12 миллисекунд, и при этом пропали очень долгие страничные отказы, которые ранее доходили до 65 миллисекунд. Таким образом, удалось получить повышение производительности системы до 20 %.

Литература

1. Вязигин С. В. Построение универсального комбинаторного пространства // Умная цифровая экономика. Екатеринбург, 2022. Т. 2. № 1. С. 41–47.
2. Масыч М. А. Разработка и реализация алгоритмов повышения доступности данных в системе LUNA: вып. квалиф. работа. Новосибирск: НГУ, 2021.
3. Чжоу З. Использование виртуализации для увеличения эффективности вычислений: дисс. ... канд. техн. наук. СПб., 2020. 238 с.
4. Dasgupta U. Shared Memory Multiprocessor and Instruction Execution in Computer Architecture // Computer Architecture. India. 2020. March 04.

Научные руководители – д-р физ.-мат. наук, проф. А. Е. Дюсембаев, КазНУ им. аль-Фараби, Казахстан, д-р техн. наук, проф. В. Э. Малышкин, ИВМиМГ СО РАН, НГУ, НГТУ

Применение методов параллельного программирования к задачам с использованием случайных процессов

С. П. Глазков

Новосибирский государственный университет
s.glazkov@g.nsu.ru

В современное время наука достигла такого момента, когда приходится решать невероятно сложные задачи с огромным количеством данных. И даже несмотря на использование самых мощных вычислительных машин многие задачи либо решались за недопустимо большое время, либо требовали слишком много памяти для входных данных. Чтобы решить эти проблемы, в сфере информационных технологий происходит массовый переход с последовательного программирования на параллельное, когда разные части задачи решаются разными вычислительными устройствами.

В настоящей работе будут рассмотрены три задачи возрастающей сложности, связанные со случайными процессами, – в частности, с гауссовским процессом и двойным процессом Пуассона, а также варианты ускорения решения этих задач с помощью параллельного программирования.

Первая задача – численное нахождение корреляционной функции гауссовского процесса $u(t)$ с известной аналитической корреляционной функцией $B(\tau)$. Для решения данной задачи используется спектральное разложение $B(\tau)$ в виде ряда Фурье [1], по которому моделируется процесс, после чего находится корреляция на сетке, положенной на некоторый интервал.

Вторая задача – нахождение корреляционной функции двойного процесса Пуассона с логнормальной интенсивностью. Для этой задачи сначала моделируется логнормальный процесс $\lambda(t) = \exp\{u(t)\}$ для гауссовского процесса $u(t)$. После этого моделируется неоднородный процесс Пуассона по алгоритму «прореживания» (англ. *thinning process* [2]), напоминающему метод исключения. Численная корреляционная функция находится аналогичным образом. Более того, по свойству двойного процесса Пуассона его корреляция в точности совпадает с корреляцией его функции интенсивности, т. е. логнормального процесса.

Третья задача связана с применением двойного процесса Пуассона для моделирования корреляционной функции смещения решетки кристалла под воздействием дислокаций. Дислокации – точно расположенные источники нарушения структуры кристалла, и для их моделирования используются точечные процессы, описывающие их расположение в слое. Таким образом, рассматривая одномерный случай, положим точки двойного процесса Пуассона координатами положения дислокаций, после

чего возможно изучение их воздействия на решетку с помощью функции смещения, которая также является случайным процессом, а значит, возможно и численное вычисление ее корреляции. Эта задача является наиболее сложной, так как для моделирования процесса смещение необходимо получить двойной процесс Пуассона, для которого, в свою очередь, необходимо получить гауссовский процесс.

Так как все три корреляционные функции получаются путем усреднения результатов на ансамбле независимых траекторий, к этим задачам очень хорошо применимы методы параллельного программирования посредством разделения исходного количества траекторий между исполнительными устройствами (потоками или процессами) с последующим суммированием результатов. Для параллельной реализации приведенных задач использовались библиотеки OpenMP и MPI. Выполнено сравнение скорости выполнения последовательной и параллельных программ. В результате тестирования оказалось, что, как правило, MPI выигрывает по времени, так как OpenMP гораздо хуже работает с логическими ядрами: заметно падение ускорения при количестве потоков большем, чем количество физических ядер. Это особенно сказывалось на более простых задачах, в частности, для первой задачи время выполнения даже увеличилось при превышении количества физических ядер. Однако с усложнением задачи ускорение перестает падать и выходит на уровень MPI. В свою очередь, библиотека MPI показывала стабильный результат с хорошим ускорением – для 12 процессов время выполнения программы для каждой задачи уменьшилось чуть больше, чем в 8 раз.

Данная работа выполнена при поддержке гранта РФФИ № 19-11-00019.

Литература

1. Chen Y. Thinning algorithms for simulating point processes. Tallahassee, 2016.
2. Sabelfeld, K. Expansion of random boundary excitations for elliptic PDEs. 2008. Vol. 13. № 5–6. P. 405–453.

Научный руководитель – д-р физ.-мат. наук, проф. К. К. Сабельфельд,
ИВМиМГ СО РАН

Разработка клеточно-автоматной модели разделения фаз с целочисленным алфавитом состояний и её параллельная программная реализация

К. А. Глинский

Сибирский государственный университет
телекоммуникаций и информатики
kirill_glinskij@mail.ru

Работа посвящена разработке клеточно-автоматной модели разделения фаз с целочисленным алфавитом состояний. За основу новой модели взята существующая клеточно-автоматная модель разделения фаз с бинарным алфавитом [1]. Модель реализована на языке Си [2] в виде функции с использованием библиотеки клеточно-автоматных топологий [3]. Функция переходов клетки автоматически применяется ко всем клеткам массива средствами библиотеки. Для ускорения тестирования модели была создана параллельная версия модуля обхода клеточного массива из библиотеки клеточно-автоматных топологий в системе с общей памятью с помощью OpenMP [4].

Новая модель имеет целочисленный алфавит, состояние её клеток $X \in \{0, 1, \dots, X_{\max}\}$ отражает количество вещества в клетке. Функция переходов клетки в новой модели задаётся следующими правилами.

1. Количество вещества в клетке, при котором действует равновесие силы тяжести и поверхностного натяжения, выбирается произвольно и принимается равным K .
2. Если состояние клетки $X > K$, то она оставляет у себя количество вещества $X' = K$ и распределяет между выбранными соседями поровну величину $X - K$. Соседи выбираются из тех, состояние которых $X_i < K$ таким образом, чтобы сумма величин $X - K$ и $\sum X_i$ выбираемых соседей, деленная на количество выбираемых соседей, была не меньше X_i каждого выбранного соседа. При этом X_i каждого выбранного соседа не должно превосходить X_j любого невыбранного соседа и количество выбранных соседей должно быть максимально возможным. Остаток от деления отдаётся случайному из выбранных соседей.
3. Если состояние клетки $X = K$ и она имеет менее трёх соседей с состояниями $X_i \geq K$, то эта клетка выбирает соседа, состояние X_j которого максимально среди всех соседей с состояниями $X_j < K$ и передаёт ему величину $K - X_i$, так что состояние этого соседа становится $X_i' = K$, а у себя оставляет величину $X' = X_i$.
4. Если состояние клетки $X < K$, и у неё нет соседей с состоянием $X_i \geq K$, то она распределяет максимальную из двух величин: X и $8 * K -$

$\sum X_i$, $i = 1 \dots 8$, среди своих соседей, передавая по $K - X_i$ каждому, начиная с соседа с максимальным X_i . Тогда эта клетка перейдет в состояние X' , равное максимальной из двух величин: 0 и $X - 8 * K - \sum X_i$.

5. В остальных случаях состояния клеток остаются неизменными: $X' = X$, $X_i' = X_i$, $i = 1 \dots 8$.

Достоинствами этой модели являются сохранение массы веществ, обусловленное функцией переходов, и возможность отслеживать толщину слоя благодаря использованию целочисленного алфавита состояний клеток.

Недостаток модели состоит в том, что области моделируемого вещества не устремляются к форме круга даже при выполнении большого количества итераций.

Дальнейшее развитие модели целесообразно проводить в направлении более точного описания процесса разделения фаз на длительных временных интервалах.

Литература

1. Бандман О. Л. Клеточно-автоматные модели пространственной динамики // Системная информатика: сб. науч. тр. Новосибирск: Изд-во СО РАН, 2006. Вып. 10. С. 59–113.
2. Язык программирования С. Лекции и упражнения / пер. с англ. 6-е изд. М.: ООО «И. Д. Вильямс», 2015. 928 с.
3. Киреев С. Е. OpenMP: Краткий обзор [Электронный ресурс]. URL: https://ssd.ssc.ru/sites/default/files/content/attach/343/lecture_openmp_2015.pdf (дата обращения: 13.07.2022).
4. Лаборатория синтеза параллельных программ [Электронный ресурс] // Отчет подразделений ИВМиМГ СО РАН о выполнении плановых заданий. 2021. С. 183 – 194. URL: https://icmmg.nsc.ru/sites/default/files/pages/files/otchet_2021_dlya_sayta_1.pdf (дата обращения: 13.07.2022).

Научный руководитель – канд. техн. наук Ю. Г. Медведев, ИВМиМГ СО РАН, НГУ

Разработка и реализация алгоритмов вызова распределённых MPI-подпрограмм из LuNA-программ

А. С. Изотов

Новосибирский государственный технический университет
a.izotov2022@gmail.com

С ростом количества задач, которые требуют больших объёмов вычислений, развивались высокоуровневые системы параллельного программирования. Хотя программы, созданные с помощью таких систем, проще в разработке, зачастую они проигрывают в быстродействии программам, написанным с использованием низкоуровневых средств параллельного программирования (MPI). Примером такой системы является разрабатываемая в ИВМиМГ СО РАН система фрагментированного программирования LuNA [1].

Для частичного решения проблемы снижения быстродействия программ, исполняемых в системе LuNA, предлагается реализовать в ней возможность вызова MPI-подпрограмм, чтобы использовать высокую производительность MPI, где это критично.

Система фрагментированного программирования LuNA – задачеориентированная система параллельного программирования. Задачей в системе LuNA является фрагмент вычислений. Фрагменты вычислений потребляют и вырабатывают фрагменты данных – конечные совокупности переменных. Фрагменты вычислений и фрагменты данных перемещаются между вычислительными узлами асинхронно. Программа в системе LuNA исполняется с помощью *runtime*-системы. В свою очередь, программы, написанные с использованием MPI, реализуют концепцию процедурного программирования и являются множеством последовательных процессов, которые взаимодействуют между собой с помощью сообщений [2].

Основной проблемой при разработке подсистемы вызова MPI-подпрограмм из LuNA-программ является трудновместимость двух разных моделей вычислений. Вот почему требуется разработать интерфейс, обеспечивающий связь между MPI-подпрограммой и LuNA-программой.

MPI-подпрограмма, работающая через данный интерфейс, должна реагировать на появление новых фрагментов данных, а также её взаимодействие с LuNA-программой не должно навязывать каких-либо ресурсоёмких процессов.

Для взаимодействия между MPI-подпрограммой и LuNA-программой предлагается использовать интерфейс, реализующий концепцию событийно-ориентированного программирования, т. е. MPI-подпрограмма будет представлять собой набор обработчиков событий. Событием может

быть, например, получение фрагмента данных от *runtime*-системы. Такой интерфейс будет соответствовать вычислительной модели системы LuNA, будет простым в реализации и не потребует много ресурсов для своей работы. Также он совместим с MPI.

Для взаимодействия LuNA-программы с MPI-подпрограммой в последней требуется определить интерфейсные методы для запроса фрагментов данных из *runtime*-системы и обработчик события получения фрагмента данных.

Основными частями порядка взаимодействия *runtime*-системы и MPI-подпрограммы через разработанный интерфейс являются следующие.

1. Вызов MPI-подпрограммы из программы на языке LuNA.
2. Передача информации о фрагментах данных, используемых в подпрограмме.
3. Запрос входных фрагментов данных на каждом узле.
4. Обработка фрагментов данных.
5. Передача результата на определённый узел.

Интерфейс был реализован в виде нового класса в исходном коде системы LuNA, также поддержка этого интерфейса была реализована в трансляторе и *runtime*-системе. Был дополнен синтаксис языка LuNA.

Для проверки работоспособности и быстродействия разработанного решения была разработана LuNA-программа умножения матриц, и её часть – редукция, реализована в двух вариантах: в виде LuNA-подпрограммы и MPI-подпрограммы.

Результаты тестирования показали, что разработанное решение работает правильно, а также что, несмотря на увеличение накладных расходов, связанных с применением разработанного интерфейса, для определённого класса задач выигрыш в производительности MPI-подпрограмм перевешивает эти накладные расходы. Таким образом, данный интерфейс может быть использован для повышения производительности LuNA-программы.

В результате была разработана подсистема вызова MPI-подпрограмм из LuNA-программ.

Литература

1. Описание языка LuNA. Описание языка LuNA на официальном репозитории системы [Электронный ресурс]. URL: https://gitlab.ssd.ssc.ru/luna/luna5/-/wikis/luna_lang_v01 (Дата обращения 13.06.2022).
2. MPI Documents. Официальная спецификация MPI [Электронный ресурс]. URL: <https://www.mpi-forum.org/docs/> (Дата обращения: 15.06.2022).

Научный руководитель – В. А. Перепёлкин, ИВМиМГ СО РАН, НГУ, НГТУ

Разработка эффективных параллельных/распределенных алгоритмов решения задачи неравновесной фильтрации (композиционная модель)

Н. М. Касымбек

Казахский национальный университет им. Аль-Фараби, г. Алматы,
Казахстан

nuryslam.qassymbek@gmail.com

Для стран с экономикой, направленной на нефтедобычу, к которым относится Казахстан, важной задачей является моделирование течения жидкости в пласте. Для более точного моделирования сложных гидродинамических процессов, происходящих в нефтяных пластах, необходимо учитывать компонентный состав фаз. Учет конвекции, диффузии, фазовых переходов и химических реакций затрудняет моделирование подобных задач. Существуют разные подходы решения данной задачи, такие как IMPES (*implicit in pressure and explicit in saturation*), SS (*simultaneous solution method*), последовательный метод (*sequential solution method*). В результате этих методов получается требующая решения система нелинейных уравнений. Решить систему таких уравнений можно методом Ньютона–Рафсона. Метод Ньютона–Рафсона линеаризует нелинейные уравнения, и после его применения получается система линейных уравнений.

Для решения систем линейных уравнений можно использовать прямые и итерационные методы. Для систем с разреженными матрицами, какой является и наша система, прямые методы могут привести к проблеме переполнения. Существуют итерационные методы, такие как CG, BiCG, GMRES.

В численных методах для получения более точного решения увеличивают количество точек разностной сетки, что в свою очередь приводит к возрастанию времени вычислений. Для ускорения вычислений используется технологии распараллеливания, как MPI, OpenMP, CUDA и др. Данные технологии требуют квалификации математика или программиста в программировании для суперкомпьютеров. В частности, программисту потребуются вручную распределять данные по процессам, описывать обмен сообщениями между процессорами, указывать передачу данных. Для избежания этих и других работ существует система LuNA. Система автоматически конструирует параллельные программы, тем самым освобождая человека от определенной части работ.

В данной работе рассматривается решение задачи моделирования многокомпонентной многофазной жидкости в пористой среде комбинированием методов Ньютона–GMRES. Разработаны параллельный алгоритм на MPI и фрагментированный алгоритм в системе LuNA метода

Ньютон–GMRES. Получены результаты вычислительных экспериментов и проведен анализ.

Научный руководитель – Д. В. Лебедев, PhD, КазНУ им. аль-Фараби,
Казахстан

Построение параллельного метода численного решения трехмерного уравнения переноса для мезомасштабной метеорологической модели

Д. В. Лещинский, А. В. Старченко, Е. А. Данилкин, С. А. Проханов

Региональный научно-образовательный математический центр
Национального исследовательского Томского государственного
университета
360flip182@gmail.com

Разнообразие в архитектуре и компоновке высокопроизводительных вычислительных систем порождает разнообразие инструментов для построения параллельных программ. На сегодняшний день в области научных вычислений самыми популярными и узнаваемыми для написания параллельных программ являются технологии MPI, OpenMP и OpenACC/CUDA. Это подталкивает исследователей к оценке перспективности использования разных подходов и их комбинации для решения своих задач.

В данной работе рассматривается параллельный алгоритм численного решения обобщенного трехмерного дифференциального уравнения конвективно-диффузионного переноса для мезомасштабной метеорологической модели. Проведено сравнение эффективности применения следующих технологий параллельного программирования: Message Passing Interface (MPI), Open Multi-Processing (OpenMP), NVidia Compute Unified Device Architecture (CUDA). Показано, что в случае использования сеток $256 \times 256 \times 32$ можно достичь ускорения в 18 (MPI/OpenMP) и 37 (CUDA) раз соответственно.

Литература

1. Bermous I., Steinle P. Efficient performance of the Met Office Unified Model v8.2 on Intel Xeon partially used nodes Geosci. Model Dev. 2015. Vol. 8. P. 769.
2. Ridwan R., et al. Performance evaluation of hybrid parallel computing for WRF model with CUDA and OpenMP // 3rd International Conference on Information and Communication Technology. Nusa Dua, 2015. P. 425–430.
3. Voronin K.V. A numerical study of an MPI/OpenMP implementation based on asynchronous threads for a three-dimensional splitting scheme in heat

transfer problems. Journal of Applied and Industrial Mathematics. 2014. Vol. 8(3). P. 436.

Реализация клеточно-автоматной модели потока ФНР на графическом ускорителе при помощи фреймворка TensorFlow

Н. А. Матолыгина, А. К. Матолыгин

Томский государственный университет, г. Томск
nat.shalyapina@gmail.com, amatolygin@mail.ru

При моделировании физических процессов при помощи клеточно-автоматных моделей важную роль играет способ их реализации. Размер клеточных пространств, который используется при моделировании, и объём требуемых вычислений наталкивают на мысль о применении высокопроизводительных вычислительных систем, в которых операции могут производиться параллельно. Написание параллельных программ – задача непростая, требующая от исследователя особых навыков параллельного программирования. Для того чтобы упростить эту задачу, в работе [1] нами был предложен специальный подход, ориентированный на применение многоядерных видеокарт и фреймворка TensorFlow [2], который эффективно эксплуатирует многоядерность видеокарт, тем самым освобождая исследователя от необходимости самостоятельно организовывать потоки данных и потоки команд. Основной структурой данных фреймворка является многомерная матрица (тензор), соответственно для того чтобы реализовать клеточно-автоматную модель при помощи тензорного подхода, необходимо сам клеточный автомат представить в виде тензора, а логику перехода из одного состояния в другое – с помощью тензорных операций. Для тензоров в TensorFlow допустимы большинство операций, которые выполняются над обычными матрицами, а также множество нетривиальных, таких как свёртка, поиск максимального / минимального элемента и т. д. Кроме этого в TensorFlow имеется возможность внедрять пользовательские операции. Фреймворк работает в связке со многими высокоуровневыми языками программирования: Python, C++, JavaScript и т. д.

В качестве примера использования предложенного подхода к реализации клеточно-автоматных моделей мы обратились к известной клеточно-автоматной модели потока ФНР [3] с одной частицей покоя. Рассматривается двумерный случай, когда поток движется вдоль ограничивающих его стенок (т. е. имитация движения по трубе). С одной стороны перпендикулярно стенкам располагается источник частиц, с другой стороны частицы поглощаются. Пространство моделирования представляет собой решётку, состоящую из клеток гексагональной формы.

Различают обычные клетки, через которые распространяется поток, клетки-источники, через которые частицы возникают в потоке, и клетки-стенки, от которых происходит отражение частиц. Каждая клетка имеет шесть соседей, и её состояние описывается булевым вектором длины 8. Первая компонента вектора отвечает за тип частицы (движущаяся или покоящаяся), следующие шесть компонент – за направления в сторону её соседей, и последняя – за тип клетки. Один такт работы автомата состоит из фазы сдвига и фазы столкновения. При сдвиге частица перемещается по направлению вектора скорости в одну из соседних клеток, а при столкновении происходит смена состояния клетки согласно правилам перехода.

Для адаптации нашего подхода необходимо:

- 1) отобразить шестиугольную форму клеток на прямоугольную без нарушения соседства;
- 2) представить клеточный автомат в виде тензора;
- 3) тензорными операциями описать возникновение и поглощение частиц;
- 4) выразить фазы сдвига и столкновения в виде тензорных операций.

На рис. 1 продемонстрирован способ отображения, в котором тип соседства чередуется в зависимости от чётности столбца.

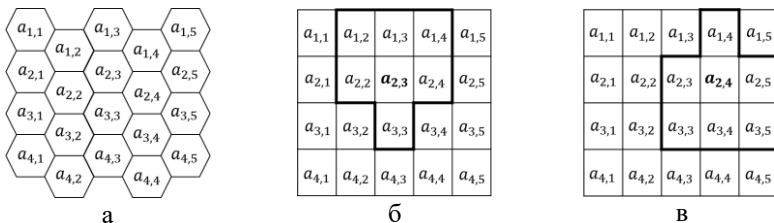


Рис. 1. Иллюстрация способа отображения шестиугольной формы клеток на прямоугольную: *а* – фрагмент с исходной формой, результат отображения и соседство; *б* – для нечётных столбцов; *в* – для чётных столбцов

Следующие этапы реализации продемонстрируем на небольшом примере. Пусть имеется некоторое начальное состояние клеточного автомата (рис. 2, *а*, серые клетки – границы). Отобразим его на прямоугольную форму клеток (рис. 2, *б*). Тензор T ранга 2, представляющий данный автомат, показан на рис. 2, *в*.

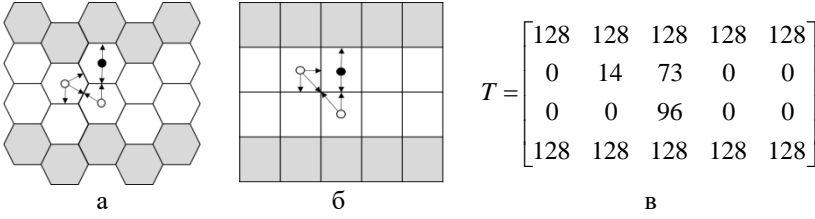


Рис. 2. Пример: a – начальное состояние автомата; b – результат отображения; v – соответствующий тензор

Реализация появления частиц осуществляется тензорной операцией побитового логического сложения тензора T и тензора S – $bitwise_ops.bitwise_or(T, S)$, где S – тензор того же размера и ранга, что и T , элементы первого столбца есть случайные целые числа в диапазоне от 0 до 128, элементы в остальных столбцах равны нулю.

Реализация поглощения частиц осуществляется тензорной операцией побитового логического умножения тензора T и тензора P $bitwise_ops.bitwise_and(T, P)$, где P – тензор того же размера и ранга, что и T , элементы во всех столбцах, кроме последнего равны 255, элементы в первой и последней строке последнего столбца равны 128, остальные 0.

$$S = \begin{bmatrix} 52 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 \\ 12 & 0 & 0 & 0 & 0 \\ 80 & 0 & 0 & 0 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 255 & 255 & 255 & 255 & 128 \\ 255 & 255 & 255 & 255 & 0 \\ 255 & 255 & 255 & 255 & 0 \\ 255 & 255 & 255 & 255 & 128 \end{bmatrix}$$

В качестве тензорной операции, которая бы могла осуществить фазу сдвига, выступает двумерная побитовая свертка, которая отличается от обычной свертки тем, что операция умножения заменена на поразрядную конъюнкцию, а операция сложения на дизъюнкцию. Однако подобную свертку не удаётся реализовать с использованием имеющихся в TensorFlow операций. Поэтому фаза сдвига была реализован нами на языке CUDA C с помощью имеющегося в TensorFlow шаблона и внедрена в систему операций TensorFlow.

Фазы сдвига и столкновения реализованы одной пользовательской операцией $FHP(input1 = T, inpu2 = R)$, представляющей собой программу, написанную на языке CUDA C, которая оперирует с данными, как с обычными двумерными массивами. На вход операции кроме тензора T поступает тензор R (того же ранга и размера, что и T), элементы которого есть случайные вещественные числа в диапазоне от 0 до 1, предназначенные для определения нового состояния обычных клеток.

Один такт работы автомата реализован в виде последовательности тензорных операций: 1) появление частиц; 2) FHP; 3) поглощение.

Для того чтобы полноценно использовать нашу реализацию модели FHP, необходимо создать программу для визуализации промоделированного векторного поля скорости потока. Создание этой программы мы планируем завершить в будущих работах.

Литература

1. Шаляпина Н. А., Громов М. Л. «Жизнь» в тензорах: реализация клеточных автоматов на видеокартах // Труды Института системного программирования РАН. 2019. Т. 31. № 3. С. 217–228.
2. TensorFlow [Электронный ресурс]. URL: <https://www.tensorflow.org> (Дата обращения: 20.09.2022).
3. Frisch U., Hasslacher B., Pomeau Y. Lattice-Gas automata for Navier-Stokes equations // Phys. Rev. Lett. 1986. № 56. P. 1505.

Научный руководитель – канд. физ.-мат. наук, доцент М. Л. Громов, ТГУ

Применение вычислительных моделей для высокоуровневой организации обработки нейрофизиологических данных на суперкомпьютерах

Е. Д. Налепова

Новосибирский государственный университет

Для решения задач организации обработки нейрофизиологических данных на высокопроизводительных вычислительных системах (ВВС) от исследователя требуется квалификация в системном программировании: необходимо разрабатывать программы (в т. ч. параллельные), взаимодействовать с ВВС и системами хранения данных, организовывать процесс обработки данных и др.

Целью настоящей работы является разработка проекта и прототипа программного инструментария для высокоуровневой организации обработки исследовательских (в частности, нейрофизиологических) данных на суперкомпьютерах.

Для обеспечения высокоуровневой организации обработки нейрофизиологических данных на суперкомпьютерах в работе предложено создание инструментария, автоматизирующего разработку *мини-приложений*. Мини-приложение – это совокупность пользовательского интерфейса с программными компонентами, обеспечивающими решение задачи на ВВС. Интерфейсы позволяют пользователям задавать параметры задач и организации расчетов на ВВС и настраиваются под потребности

конкретной группы нейрофизиологов. В основе автоматизации лежит база знаний о расчетах в предметной области. Предложено формальное представление знаний об обработке данных в нейрофизиологии в виде вычислительных моделей (ВМ), являющихся ориентированными двудольными графами, в которых вершинами обозначаются переменные и вычислительные операции [1, 2]. ВМ позволяют конструировать, обобщать, переиспользовать, адаптировать сценарии организации вычислений в предметных областях. Требования к генерации мини-приложения для конкретной задачи формализуются спецификацией задачи на ВМ, являющейся совокупностью заданных на графе множеств входных и выходных переменных и нефункциональных требований к выводу алгоритма получения значений выходных переменных по значениям входных.

Реализован прототип программного инструментария – системные компоненты, обеспечивающие высокоуровневое решение задач обработки исследовательских данных, вычислительная модель, формализующая одну из задач обработки нейрофизиологических данных, и мини-приложение, которое может быть сгенерировано по спецификации задачи на ВМ и предназначено для решения прикладным специалистом задачи обработки данных, собранных с электроэнцефалографов, в терминах его предметной области.

Литература

1. Вальковский В. А., Малышкин В. Э. Синтез параллельных программ и систем на вычислительных моделях. Новосибирск: Наука. Сиб. отделение, 1988. 129 с.
2. Malyshkin V. Active Knowledge, LuNA and Literacy for Oncoming Centuries // LNCS. 2015. Vol. 9465. P. 292–303.

Научный руководитель – М. А. Городничев, ИВМиМГ СО РАН, НГУ, НГТУ

Разработка и реализация алгоритмов динамической балансировки вычислительной нагрузки для подсистемы воспроизведения трасс в системе LuNA

М. П. Саяпин

Новосибирский государственный технический университет
mathew0505@mail.ru

Разработка высокопроизводительных параллельных программ для распределенных систем часто является сложной задачей из-за

необходимости декомпозиции данных и вычислений. Востребованное направление в сфере параллельного программирования – разработка систем автоматического построения распределенных программ. Разработка таких средств является сложной задачей в общем случае, что зачастую обуславливает меньшую эффективность конструируемых программ. Одним из таких средств программирования является разрабатываемая в ИВМиМГ СО РАН система фрагментированного программирования LuNA [1].

Для возможного решения упомянутой проблемы более низкой эффективности программы, исполняемой в данной системе, был реализован подход оптимизации автоматического построения параллельных программ – это оптимизация на основе трассировки [2].

Этот подход предполагает, что программа сначала запускается на некоторых характерных входных данных и ее ход работы записывается в файл, называемый трассой. Трасса анализируется и воспроизводится с помощью специальной легковесной исполнительской системы, называемой проигрывателем трасс, для воспроизведения хода событий обычной программы [2].

Однако такой подход не всегда выгоден, потому что при воспроизведении трассы или изменении входных данных может возникать дисбаланс вычислительной загрузки.

Была поставлена задача добавить в проигрыватель трасс динамический балансировщик вычислительной загрузки, который будет отслеживать уровень вычислительной загрузки узлов и при выявлении дисбаланса загрузки между узлами устранять его за счет перераспределения нагрузки.

Система фрагментированного программирования LuNA – задачеориентированная система параллельного программирования. Задачами в системе LuNA являются фрагменты вычислений – программные единицы, содержащие описание входных, выходных фрагментов данных и исполняемый код. Фрагменты вычислений потребляют и вырабатывают фрагменты данных – набор переменных.

Плюсами действующей подсистемы воспроизведения трасс является отсутствие сборщика мусора и малое количество накладных расходов.

Минусом является возможный динамический дисбаланс вычислительной загрузки при изменении конфигурации оборудования или входных данных.

При выборе алгоритма динамической балансировки вычислительной нагрузки были поставлены следующие требования: малое количество накладных расходов алгоритма, параметризация алгоритма, а также улучшение эффективности программ при увеличении вычислительных мощностей. Был выбран алгоритм динамической балансировки WorkStealing [3].

При интеграции алгоритма динамической балансировки в подсистему воспроизведения трасс были рассмотрены следующие вопросы:

- 1) обнаружение дисбаланса загрузки;
- 2) вычисление загрузки узла;
- 3) переброска фрагментов данных;
- 4) миграция фрагментов вычислений между узлами;
- 5) остановка системы.

Был переработан класс считывания трассы, добавлены методы динамической балансировки, переработан обработчик событий, переработана остановка работы системы, а также главная функция воспроизведения трассы потоками.

Для тестирования было выбрано матричное умножение. Результаты проведенных тестирований указывают на небольшие накладные расходы динамического балансировщика вычислительной нагрузки в подсистеме воспроизведения трасс, а также на корректность распределения динамическим балансировщиком вычислительной нагрузки по узлам.

В результате обеспечена динамическая балансировка вычислительной нагрузки в подсистеме проигрывателя трасс.

Литература

1. Описание языка LuNA [Электронный ресурс]. URL: https://gitlab.ssd.sccc.ru/luna/luna5/-/wikis/luna_lang_v01 (Дата обращения 13.06.2022).
2. Cederman D., Tsigas Ph. Dynamic Load Balancing Using Work-Stealing // GPU Computing Gems Jade Edition. 2011. P. 485–500.
3. Malyshkin V. E., Perepelkin V. A. Trace-based optimization of fragmented programs execution in LuNA system // Lecture Notes in Computer Science. 2021. Vol. 12942. P. 3–10.

Научный руководитель – В. А. Перепёлкин, ИВМиМГ СО РАН, НГУ,
НГТУ

Параллельная реализация задачи фильтрации двухфазной жидкости

В. А. Спирин

Новосибирский государственный университет
v.spirin@g.nsu.ru

В области численного моделирования стоит проблема разработки эффективных (по времени выполнения, расходу памяти, нагрузке на процессор) параллельных программ. Для её решения имеет смысл рассмотреть автоматизацию конструирования параллельных программ [1],

так как соответствующие системы в принципе способны эффективно параллельно реализовывать задачи подобного типа.

В работе исследуется одна из таких задач: моделирование процесса фильтрации двухфазной жидкости на основе законов сохранения в интегральной форме [2]. Основной целью является создание такой эффективной параллельной программы, которая будет соответствовать модели вычислений системы [3], которая на данном этапе находится в разработке. Поэтому имеет смысл создать параллельную программу вручную, и при достижении ею высокой эффективности её можно считать тем желаемым результатом, который должен получаться при автоматическом конструировании.

В работе были поставлены следующие задачи: анализ существующих реализаций [4]; построение структурной схемы зависимостей по данным; создание эффективной фрагментированной параллельной программы.

В результате анализа реализаций было выявлено наиболее времязатратное место – подпрограмма, в которой применяется метод сопряжённых градиентов. Она использует несколько основных математических операций – векторные и матрично-векторные. Параллельная реализация векторных операций достаточно прямолинейна. В свою очередь, матрично-векторные операции могут реализовываться с помощью разных параллельных алгоритмов: тут есть зависимость от того формата представления матриц и векторов, которую предлагает программа (например, разреженный формат матриц).

Все матрично-векторные операции данной задачи принадлежат трём типам по особенностям параллельной реализации: 1) решение системы линейных алгебраических уравнений методом прогонки (умножение обратной матрицы на вектор); 2) умножение матрицы на вектор; 3) умножение транспонированной матрицы на вектор.

Для тестирования нефункциональных свойств разрабатываемых соответствующих подпрограмм была обеспечена возможность их изолированного запуска, разработан стенд. Для подачи в него входных данных используется сериализация и десериализация векторов и матриц для получения результата.

Параллельная реализация метода прогонки (1) происходит по следующему принципу: так как в данном численном алгоритме матрица должна быть симметричной и блочно-трёхдиагональной, то допускается применение алгоритма к каждому блоку матрицы независимо от других. Также была сконструирована параллельная версия операции умножения матрицы на вектор (2).

Реализация параллельной версии операции умножения транспонированной матрицы на вектор (3) требует специального обхода матрицы, из-за чего достичь ускорения можно, что было подтверждено на

тестах, за счёт увеличения используемой памяти. А именно, результирующий вектор вычислялся частично каждым потоком в отдельном буфере с последующей редукцией частичных результатов.

В итоге решены следующие задачи: проанализированы исходные коды решения задачи, создана структурная схема зависимостей по данным, разработаны требуемые параллельные версии матрично-векторных операций. В дальнейшем планируется использовать этот опыт для повышения эффективности конструируемых программ в системе LuNA.

Литература

1. Иванов М. И., Кремер И. А., Лаевский Ю. М. Моделирование процесса фильтрации двухфазной жидкости на основе законов сохранения в интегральной форме. Екатеринбург, 2020. 35 с.
2. Перепёлкин В. А., Иванов М. И. Повышение производительности LuNA-программ на основе воспроизведения трасс // Десятая Сибирская конференция по параллельным и высокопроизводительным вычислениям / под ред. А. В. Старченко. Томск, 2021. С. 29–36.
3. Перепелкин В. А., Софронов И. В., Ткачёва А. А. Автоматизация конструирования численных параллельных программ с заданными нефункциональными свойствами на базе вычислительных моделей // Проблемы информатики. Новосибирск, 2017. № 4. С. 47–60.
4. Malyshkin V. E., Perepelkin V. A. LuNA Fragmented Programming System, Main Functions and Peculiarities of Run-Time Subsystem // Proceedings of the 11th International Conference on Parallel Computing Technologies, LNCS. 2011. Vol. 6873. P. 53–61.

Научный руководитель – В. А. Перепёлкин, ИВМиМГ СО РАН, НГУ

Оптимизация алгоритма работы клеточного автомата «Домино» и его реализация с помощью библиотеки клеточно-автоматных топологий

Ю. С. Трубицына

Новосибирский государственный университет
y.trubitsyna@g.nsu.ru

Клеточно-автоматное моделирование является мощным инструментом для исследования поведения сложных систем. При увеличении размеров клеточных массивов в таких моделях последовательная программная реализация клеточных автоматов требует недопустимо большого времени для исполнения. Следовательно, довольно часто возникает задача распараллеливания программных реализаций таких моделей.

Работа посвящена развитию асинхронного клеточного автомата «Домино» [1] в направлении параллельной программной реализации. По причине того, что задача распараллеливания асинхронной реализации является трудновыполнимой, была проделана работа по переходу от асинхронной реализации клеточно-автоматной модели к синхронной.

В рамках данной работы рассматривается реализация синхронной клеточно-автоматной модели «Домино» с помощью библиотеки клеточно-автоматных топологий [2], а также ее оптимизация. Выбор такого способа реализации обусловлен тем, что библиотека позволит получить параллельный код автоматически.

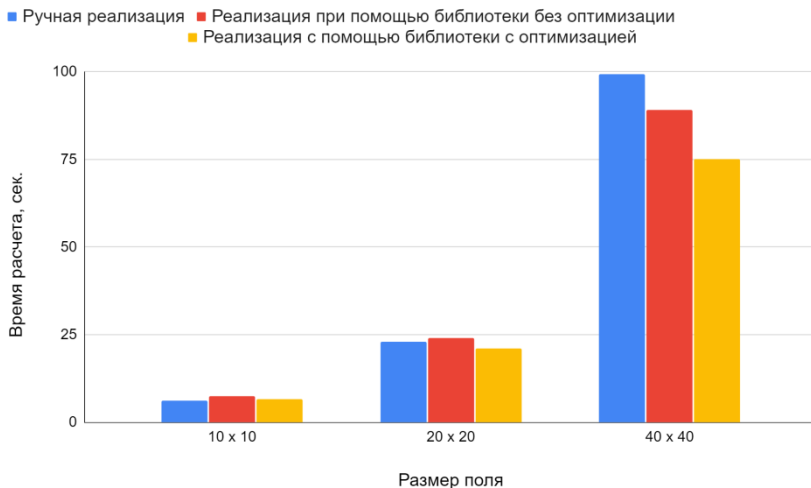
На начальном этапе была проведена работа над библиотекой, а именно реализована недостающая топология¹ (2D размером 5x5, клетки квадратные, 24 соседа), необходимая для модели. Далее были реализованы три программных модуля – препроцессор, симулятор и постпроцессор – необходимых для реализации клеточно-автоматной модели.

В препроцессоре производится инициализация управляющих структур, таких как размер клеточного массива, размер памяти, выделяемый под хранение состояния клетки, а также задается используемая топология и инициализируется клеточный массив. После работы препроцессора исполняется симулятор, который применяет функцию переходов ко всем клеткам модели заданное количество итераций. Постпроцессор выводит на экран выходные данные, полученные в результате работы симулятора.

Функция переходов, используемая в модели из [1], основана на результатах сопоставления окрестности выбранной клетки с 24 шаблонами. В программной реализации такое сравнение приводит к многократному перебору шаблонов. Поэтому была проведена оптимизация функции переходов клеточного автомата путем объединения всех 24 шаблонов в один мультишаблон. Идея оптимизации заключается в том, чтобы каждому шаблону сопоставить соответствующий двоичный разряд в булевом векторе. Выходными данными после применения мультишаблона является булев вектор, на основе которого и происходит эволюция модели. Применение такой оптимизации позволило избавиться от последовательного сравнения части клеточного массива с 24 шаблонами.

Также было проведено тестирование с целью сравнения производительности трех синхронных реализаций: ручной (без использования библиотеки), с использованием библиотеки и с использованием библиотеки и мультишаблона. Измерения производились при запуске программ на 40 000 итераций. Результаты представлены на диаграмме.

¹ Топология – способ организации связей между соседними вершинами клеточного массива.



Сравнение различных реализаций

Таким образом, была разработана программа, которая реализует алгоритм синхронной работы клеточно-автоматной модели «Домино», и выполнена оптимизация функции переходов данной модели. В результате проделанной работы время счёта при больших размерах области сократилось на 25%. Переход на использование библиотеки клеточно-автоматных топологий в дальнейшем позволит автоматически получить параллельную реализацию.

Литература

1. Лаборатория синтеза параллельных программ [Электронный ресурс] // Отчет подразделений ИВМиМГ СО РАН о выполнении плановых заданий. 2021. С. 183–194. URL: https://icmmg.nsc.ru/sites/default/files/pages/files/otchet_2021_dlya_sayta_1.pdf (Дата обращения: 13.07.2022).
2. Hoffmann R., Désérable D., Seredyński F. A cellular automata rule placing a maximal number of dominoes in the square and diamond // The Journal of Supercomputing. 2021. Vol 77. P. 9069–9087.

Научные руководители – С. Е. Киреев, ИВМиМГ СО РАН, НГУ, канд. техн. наук Ю. Г. Медведев, ИВМиМГ СО РАН, НГУ

СОДЕРЖАНИЕ

Предисловие	3
Программа лекций Всероссийской летней XXXVII молодежной школы-конференции по параллельному программированию с международным участием	5
Будущие технологии параллельного программирования. Малышкин В. Э., ИВМиМГ СО РАН, г. Новосибирск	5
Технология фрагментированного программирования и система LuNA. Перепёлкин В. А., ИВМиМГ СО РАН, г. Новосибирск	5
Прямые методы решения разреженных СЛАУ: особенности параллелизации. Соловьев С. А., ИНГГ СО РАН, г. Новосибирск	5
Высокопроизводительные вычисления на базе Desktop Grid. Ивашко Е. Е., ИПМИ КарНЦ РАН, г. Петрозаводск	6
Проблемы отладки и оптимизации параллельных программ. Методы решения. Власенко А. Ю., ИВМиМГ СО РАН, г. Новосибирск	7
Применение нейронных сетей при решении геофизических задач. Дучков А. А., ИНГГ СО РАН, г. Новосибирск	7
Решение уравнения эйконала с помощью нейронной сети для моделирования времен пробега сейсмических волн. Исследование параметров архитектуры, влияющих на устойчивость получаемого решения. Грубась С. И., ИНГГ СО РАН, г. Новосибирск	7
Численный прогноз опасных метеорологических условий с использованием суперкомпьютера. Старченко А. В. и др., ТГУ, г. Томск.....	8
Interpretable ML для анализа мультимодальных данных психометрии и нейрофизиологии. Рудыч П. Д., НИИНМ СО РАН, г. Новосибирск.....	8
Программа лекций Всероссийской летней XXXIX молодежной школы-конференции по параллельному программированию с международным участием	9
Система автоматического конструирования параллельных программ LuNA. Перепёлкин В. А., ИВМиМГ СО РАН, г. Новосибирск	9
Новое оборудование ССКЦ СО РАН для обучения нейросетевых моделей. Городничев М. А., ИВМиМГ СО РАН, г. Новосибирск	9

Дискретные методы в задачах моделирования пространственной динамики природных процессов. Медведев Ю. Г., ИВМиМГ СО РАН, г. Новосибирск	9
Тезисы докладов участников	10
Абрамушкина Е. С. Проектирование подсистемы профилирования для системы фрагментированного программирования LuNA	10
Аксюк И. А. Случайный сферический процесс дрефта-диффузии. Сравнение OpenMP и MPI подходов	12
Баранов И. Н. Разработка экспериментальной библиотеки работы с распределенным массивом для поддержки автоматического конструирования параллельных программ	13
Бухашеев О. В. Распараллеливание с помощью OpenMP и MPI глобального стохастического алгоритма блуждания по сетке для решения нелинейной системы уравнений Бюргерса	14
Валитов А. А. Управление распределенными фрагментированными данными в системе LuNA	16
Вязигин С. В. Использование оперативной памяти для кеширования в системах с распределенной виртуальной памятью	18
Глазков С. П. Применение методов параллельного программирования к задачам с использованием случайных процессов	21
Глинский К. А. Разработка клеточно-автоматной модели разделения фаз с целочисленным алфавитом состояний и её параллельная программная реализация	23
Изотов А. С. Разработка и реализация алгоритмов вызова распределённых MPI-подпрограмм из LuNA-программ	25
Касымбек Н. М. Разработка эффективных параллельных / распределенных алгоритмов решения задачи неравновесной фильтрации (композиционная модель)	27
Лещинский Д. В., Старченко А. В., Данилкин Е. А., Проханов С. А. Построение параллельного метода численного решения трехмерного уравнения переноса для мезомасштабной метеорологической модели	28
Матолыгина Н. А., Матолыгин А. К. Реализация клеточно-автоматной модели потока ГНР на графическом ускорителе при помощи фреймворка TensorFlow	29

Налепова Е. Д. Применение вычислительных моделей для высокоуровневой организации обработки нейрофизиологических данных на суперкомпьютерах	32
Саяпин М. П. Разработка и реализация алгоритмов динамической балансировки вычислительной нагрузки для подсистемы воспроизведения трасс в системе LuNA	33
Спирин В. А. Параллельная реализация задачи фильтрации двухфазной жидкости	35
Трубицына Ю. С. Оптимизация алгоритма работы клеточного автомата «Домино» и его реализация с помощью библиотеки клеточно-автоматных топологий	37

Научное издание

Всероссийская летняя XXXVII молодежная Школа-конференция по
параллельному программированию с международным участием
(г. Новосибирск, 5 – 16 июля 2021 г.)

Всероссийская летняя XXXIX молодежная Школа-конференция по
параллельному программированию с международным участием
(г. Новосибирск, 4 – 15 июля 2022 г.)

Тезисы докладов

Редактор *С. В. Исакова*

Обложка *Е. В. Неклюдовой*

Подписано к публикации 22.05.2023 г.

Формат 60 x 84/16. Уч.-изд. л. 2,75. Усл. печ. л. 2,6.

Заказ № 126

Издательско-полиграфический центр НГУ
630090, Новосибирск, ул. Пирогова, 2.

ISBN 978-5-4437-1460-8



9 785443 714608