Исследование масштабируемости параллельных алгоритмов методом агентно-ориентированного моделирования*

Б.М. Глинский, А.С. Родионов, М.А. Марченко, Д.А. Караваев,
Д.И. Подкорытов, Д.В. Винс

Федеральное государственное бюджетное учреждение науки
Институт вычислительной математики и математической геофизики
Сибирского отделения Российской академии наук

В докладе исследуется возможность масштабирования алгоритмов распределенного статистического моделирования и сеточных методов на большое число (сотни тысяч и миллионы) вычислительных ядер предполагаемого экзафлопсного суперкомпьютера. Актуальность предмета исследования обосновывается оценкой вычислительной эффективности этих алгоритмов в свете ожидаемого появления к 2020 году суперкомпьютеров экзафлопсного уровня производительности. Рассматриваются особенности поведения вычислительных алгоритмов при их реализации на кластерах Сибирского суперкомпьютерного центра (ССКЦ) и оценка их масштабируемости на предполагаемых архитектурах MPP и гибридной, проведённая с помощью мультиагентной системы моделирования AGNES (AGent NEtwork Simulator).

1. Введение

Детальный анализ масштабируемости параллельных алгоритмов в сложных системах большой размерности, к которым, в частности, относятся будущие суперкомпьютеры экзафлопсного уровня производительности, невозможен применением одних лишь математических моделей. Повышение параллелизма является одной из основных тенденций развития суперЭВМ.

Так, например, недавний лидер мирового списка Top 500 Sequoia это суперкомпьютер семейства IBM BlueGene/Q построен на шестнадцатиядерных процессорах Power BQC, всего 1572864 вычислительных ядер, производительность на Linpack 16.32 PFLOPS. Для эффективного использования таких вычислительных ресурсов необходимо разрабатывать прикладные алгоритмы и программы с высокой степенью параллелизма. Проблемы прикладного программного обеспечения будут обостряться с переходом на суперЭВМ экзафлопной производительности и уже сейчас необходимо разрабатывать алгоритмы и программы с высокой степенью параллелизма и уметь оценивать их поведение на десятках и сотнях миллионов вычислительных ядер.

Сложный характер алгоритмов и стохастическая природа входящих потоков данных и задач делает необходимым проведение имитационного моделирования таких систем. С другой стороны, размерность детализированных моделей такова (сотни тысяч и миллионы одновременно присутствующих в модели объектов), что использование традиционных систем имитационного моделирования становится невозможным как для описания моделей, так и их исполнения.

Системы имитационного моделирования, в сравнении с универсальными языками программирования, дают ряд преимуществ. Они обеспечивают исследователя естественной средой для создания имитационных моделей и предоставляют такие возможности, как генерирование случайных чисел с заданным распределением вероятности, продвижение модельного времени, обработка списков текущих и будущих событий и др.; имеют встроенные механизмы представления параллельных процессов. Имитационные модели, созданные с помощью систем моделирования, как правило, проще модифицировать и использовать.

* *Контактная информация: 8(383)3306279 www.sscc.ru, www2.sscc.ru
Настоящая работа проводилась при финансовой поддержке грантов РФФИ 12-01-00034, 12-01-00727; МИП № 39 СО РАН, МИП № 47 СО РАН, МИП № 126 СО РАН, МИП № 130 СО РАН.

Отметим несколько важных моментов, связанных с исследованием масштабируемости алгоритмов:

- Исследование свойств масштабируемости параллельных алгоритмов является важной задачей при оценке эффективности их реализации, как на настоящих, так и будущих суперкомпьютерах пета- и экзафлопсного уровня.
- Данная проблема 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 можно, путем реализации их на имитационной модели, содержащей тысячи и миллионы вычислительных ядер.
- При исследовании масштабируемости абсолютные значения оцениваемого времени исполнения программ не имеют значения, важно лишь относительное его изменение при наращивании ресурсов моделируемой вычислительной системы.
- Имитационная модель позволит 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 данной работе с использованием системы имитационного моделирования AGNES (AGent NEtwork Simulator), разработанной в ИВМиМГ СО РАН рассматриваются особенности масштабирования двух типов алгоритмов: распределенного статистического моделирования и численного моделирования 3D сейсмических полей при их отображении на архитектуры экзафлопсного класса.

Моделирование исполнения параллельных алгоритмов для анализа асштабируемости проводилось как на кластере с MPP-архитектурой с пиковой производительностью 30 TFlops, так и на гибридном кластере ССКЦ, который состоит из 40 вычислительных узлов HP SL390s G7. Каждый узел содержит: два 6-ядерных CPU Xeon X5670 (2.93GHz); 96 ГБ оперативной памяти;

три карты NVIDIA Tesla M2090. Каждая карта содержит GPU с 512 ядрами и 6 ГБ оперативной памяти. Суммарно гибридный кластер содержит 80 процессоров (480 ядер) CPU и 120 процессоров (61440 ядер) GPU. Пиковая производительность – 85 TFlops, на тесте Linpak – 38 TFlops.

2. Агентно-ориентированная система имитационного моделирования AGNES

Пакет AGNES (AGent NEtwork Simulator) базируется на Java Agent Development Framework (JADE) [4]. 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. AGNES состоит из двух типов агентов [5,6]:

- управляющие агенты (УА), которые создают среду моделирования;
- функциональные агенты (ФА), которые образуют модель, работающую в среде моделирования.

Основная задача функциональных агентов – имитация работы исследуемой системы. Основные задачи управляющих агентов: инициализация и запуск модели; сбор и хранение информации о ходе моделирования; синхронизация модельного времени; перераспределение нагрузки между вычислителями, участвующих в моделировании; взаимодействие с пользователем (вывод отчетов и возможность влиять на ход моделирования); обеспечение отказоустойчивости, восстановление модели; балансировка нагрузки.

Для обеспечения отказоустойчивости AGNES реализуется несколько механизмов: отсутствие централизованного хранения данных для восстановления; хранение информации располагается частями на разных агентах среды моделирования, и эта информация хранится с избытком, для гарантии её восстановления; динамическое изменение хранилищ информации во время работы среды моделирования, т.е. в ходе моделирования хранилище данных о конкретном агенте изменяется, и эта обязанность переходит от одного агента к другому.

Таким образом, основные принципы улучшения отказоустойчивости среды моделирования: децентрализации хранилищ и избыточность информации. Эффективность средств обеспечения отказоустойчивости проверена на экспериментах с физическими нарушениями аппаратной среды (локальной сети персональных ЭВМ) при реализации исполнения модели сенсорной сети.

Приложение AGNES – это распределенная мультиагентная сеть (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.

Мультиагентный подход 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.

3. Модель экзафлопсной суперЭВМ и отображение параллельной программы на ее архитектуру

Архитектура ЭВМ экзафлопсной производительности в настоящее время не определена. Есть некоторые соображения, однако какой она будет, вероятнее всего, это мы узнаем только ближе к 2018 году. Однако, в некоторых работах зарубежных и отечественных авторов высказаны предположения, что это будут гибридные архитектуры, например в работе [7] дано обоснование необходимости применения гибридных архитектур для достижения экзафлопсной производительности. В своей работе мы предполагаем экстенсивное развитие инструментального вычислительного кластера НКС-30Т+GPU ЦКП ССКЦ СО РАН (многократное увеличение количества существующих ядер, в составе этого кластера, как было указано выше, есть и однородные структуры и гибридные). Тем самым делается оценка производительности «снизу», поскольку естественно ожидать повышения характеристик ядер и интерконнекторов ЭВМ экзафлопсной производительности по сравнению с существующими.

Архитектура вычислительной системы как таковая явно в модели не присутствует, поскольку значение имеют лишь задержки на вычисления и обмены. Соответственно этому, задержки либо (как в случае представленных экспериментов) замеряются в ходе проведения реальных расчётов на конкретных ВС, либо задаются как функция от вложения графа задачи и графа размещения данных в моделируемую ВС. При этом функция вычисления задержки на передачу данных может быть достаточно сложна. Делается допущение, что данные передаются по кратчайшим путям. Возможные дополнительные задержки моделируются добавлением случайной составляющей.

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.

Далее рассмотрим применение системы AGNES для исследования масштабируемости распределенного статистического моделирования и численного моделирования распространения сейсмических полей в 3D неоднородных упругих средах.

4. Имитационное моделирование метода Монте-Карло.

С целью изучения возможности масштабирования распределенного статистического моделирования на большое число вычислительных ядер производилось имитационное моделирование работы экзафлопсного суперкомпьютера при решении задачи динамики разреженного газа по методу прямого статистического моделирования (ПСМ). Это типичная задача статистического моделирования, требующая всей вычислительной мощи многопроцессорного суперкомпьютера, т.е. требующие моделирования экстремально большого количества независимых реализаций [8]. К числу таких проблем относятся задачи моделирования с использованием ПСМ течений разреженного газа с учетом химических реакций, задачи переноса излучения и теории дисперсных систем.

Общая схема вычислений по методу Монте-Карло (см. рис.1):

Шаг 1: Подготовка к моделированию независимых реализаций на группах ядер.
Шаг 2: Моделирование реализаций, вычисление выборочных средних для группы.
Шаг 3: Сбор и осреднение данных.

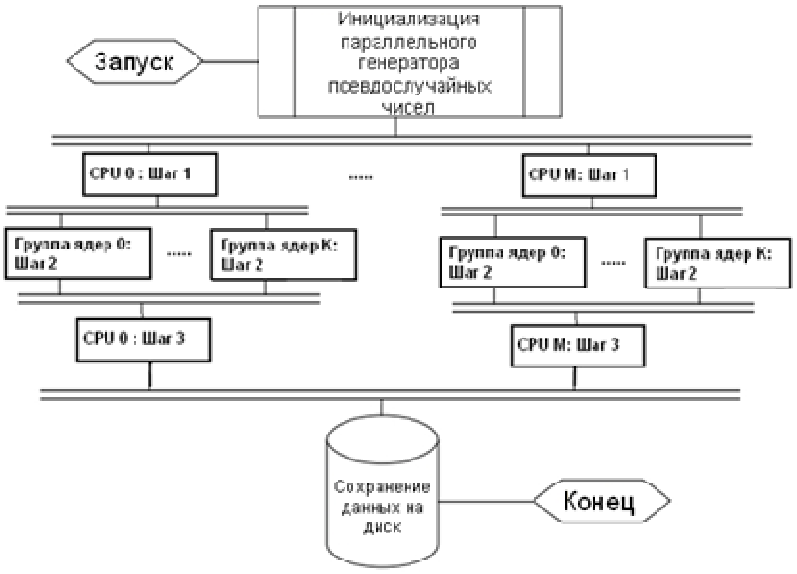

Рис. 1. Схема параллельных вычислений методов Монте-Карло.

Имитационное моделирование проводилось с использованием мультиагентной системы AGNES. Для имитации вычислений методов Монте-Карло созданы два класса функциональных агентов:

- *DataAgregator*: ядро-сборщик, собирает информацию о вычислениях, обрабатывает и агрегирует её. Возможно иерархическое построение сборщиков, которые на нижнем уровне обрабатывают данные непосредственно 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 независимых реализаций методов Монте-Карло на нескольких вычислительных узлах, группа ядер-вычислителей. 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 [9]. Область применения библиотеки: «большие» задачи статистического моделирования в естественных и гуманитарных науках (физика, химия, биология, медицина, экономика и финансы, социология и др.) Библиотека PARMONC установлена на кластерах Сибирского суперкомпьютерного центра (ЦКП ССКЦ СО РАН) и может использоваться на вычислительных системах с аналогичной архитектурой. При этом использование библиотеки

не привязано к каким-то определенным компиляторам языков C и FORTRAN или MPI. Инструкции по использованию библиотеки с примерами можно найти по ссылкам [10].

Как известно, теоретическое ускорение при распараллеливании для методов статистического моделирования практически "идеальное", что подтверждается численными расчетами при числе вычислительных ядер порядка нескольких тысяч [11]. Тем не менее, при числе ядер порядка сотен тысяч или нескольких миллионов вопросы организации счета требуют серьезного исследования, поскольку при этом возникают проблемы с большой загрузкой ядер-сборщиков, которые периодически собирают статистику с ядер-вычислителей. А именно, проведенное имитационное моделирование показало, что при большом числе используемых вычислительных ядер (больше 10000) реальное ускорение от распараллеливания существенно отличается от теоретического, что связано с большой загрузкой выделенных ядер-сборщиков, которые обрабатывают поступающие пакеты данных с ядер-вычислителей. При этом до 1000 ядер ускорение в модели совпадает с ускорением в реальных расчетах. С целью повышения эффективности распараллеливания исследовались различные варианты организации обмена данными между ядрами.

А именно, целесообразно осуществлять периодическую пересылку результатов промежуточного осреднения реализаций, независимо полученных на загруженных ядрах (ядрах-вычислителях), на выделенные ядра (ядра-сборщики), объединенные в многоуровневую структуру. Ядра-сборщики будут периодически получать переданные им данные и осреднять их, передавая затем результаты на ядро (с номером 0), соответствующее вершине многоуровневой структуры (см. рис. 2 и 3). Будем называть такое ядро *главным ядром-сборщиком*; в числе его задач – сохранение осредненных данных на диск. Рассчитанные на главном ядре-сборщике осредненные значения будут соответствовать выборке, полученной совокупно на всех ядрах-вычислителях. Распределенное статистическое моделирование на разных вычислительных ядрах-вычислителях производится в асинхронном режиме. Отправка и получение результатов статистического моделирования также осуществляется в асинхронном режиме [12].

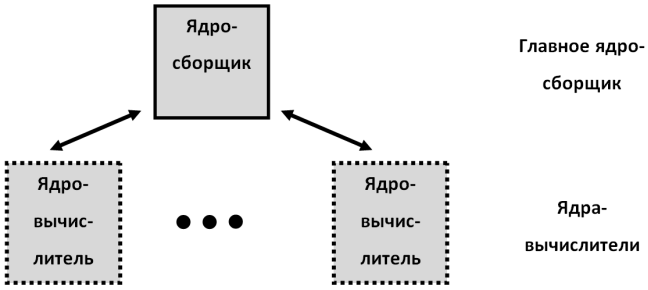

Рис. 2. Вариант организации связей между ядрами: одно главное ядро-сборщик.

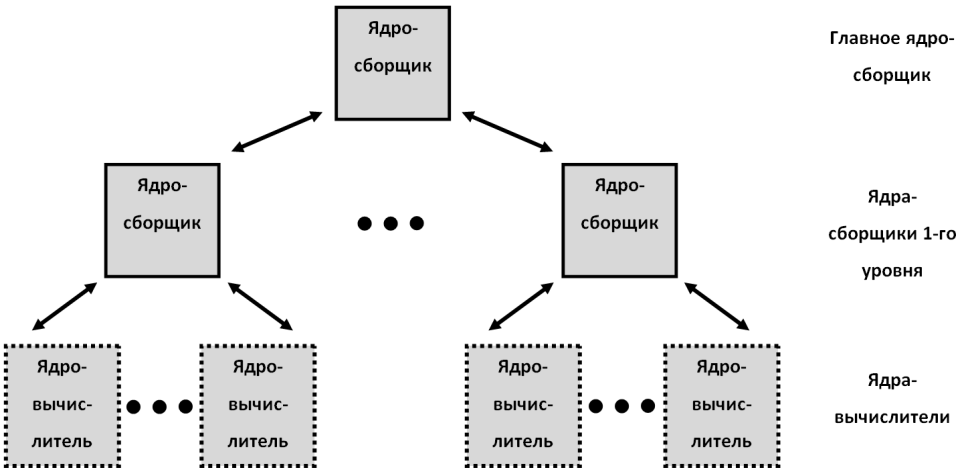

Рис. 3. Вариант организации связей между ядрами: один уровень промежуточных ядер-сборщиков.

На кластере НКС-30Т Сибирского суперкомпьютерного центра с использованием библиотеки PARMONC, был произведен ряд расчетов для общего числа ядер, примерно равного 1000. 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. По результатам расчетов был сделан вывод, что требуемый уровень относительной статистической погрешности в 0.1% достигается при объеме выборки равном L = 240 000. Среднее время моделирования одной реализации - примерно 12 сек. Для ядер-вычислителей обмен данными с главным ядром-сборщиком происходил после каждой смоделированной на них реализации.

Отображение модели на вычислительный кластер выглядит следующим образом: на каждом сервере запускается JVM и отдельный контейнер JADE. На каждом контейнере запускаются агенты имитирующие расчет методов Монте-Карло. Каждый агент *MonteCarlo* содержит внутри себя группу циклических поведений, каждое из которых имитирует расчет независимых реализаций на одном вычислительном узле. Подобное представление позволяет моделировать расчеты по методу Монте-Карло на 10^7 вычислительных узлах с использованием 10^4 агентов (см. рис. 4).

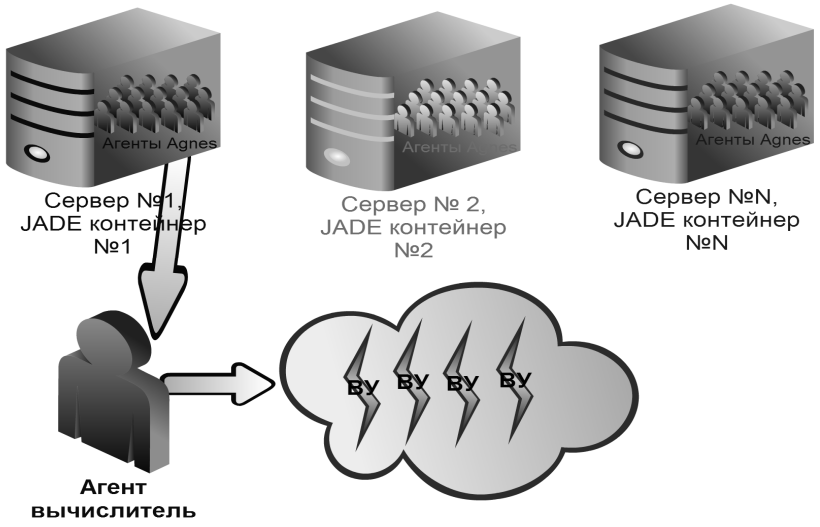

Рис. 4. Схема отображения модели на вычислительный кластер.

5. Результаты расчетов на имитационной модели для методов Монте-Карло

При имитационном моделировании расчетов по методу Монте-Карло за основу взята MPP-архитектура кластера НКС-30Т. При этом исследовалась величина относительного ускорения от распараллеливания при расчётах на M ядрах, определенная следующим образом:

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

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

В первой серии экспериментов рассматривались два варианта организации обмена данными между ядрами-вычислителями и главным ядром-сборщиком:

- с использованием одного уровня промежуточных ядер-сборщиков (см. рис. 3).
- без использования промежуточных ядер-сборщиков (см. рис. 2);

В первом варианте ядра-вычислители были поделены на M_1 равных частей (M_1 = 10, 20, 100), для каждой из которых данные с ядер-вычислителей всегда отправлялись на «свое» ядро-сборщик. В свою очередь, M_1 ядер-сборщиков отправляли данные на главное ядро-сборщик. Во втором варианте для определенности будем считать, что параметр $M_1 = 0$.

На рис.5 приведена зависимость относительного ускорения $S_L(M)$ от общего числа моделируемых ядер M. Моделирование проводилось до $M = 5*10^5$ ядер, но для показательности на рисунке приведены данные до $M = 10^5$. На рисунке ясно видна закономерность: увеличение числа ядер-сборщиков приводит к увеличению относительного ускорения.

В этой связи интересно исследовать вопрос об оптимальном (в смысле максимального значения относительного ускорения) числе ядер-сборщиков при фиксированном общем числе моделируемых ядер.

Во второй серии экспериментов при фиксированном числе моделируемых ядер $M = 10^6$ варьировалось число ядер-сборщиков M_1, а также соответствующее число ядер-вычислителей в каждой группе, связанной со «своим» ядром-сборщиком. На рис. 6 приведен график зависимости относительного ускорения S_L от числа ядер-сборщиков M_1. Из рисунка видно, что максимальная величина относительного ускорения достигается при M_1, приближенно равном 1000. Это говорит о том, что при меньшем значении M_1 ядра-сборщики перегружены обработкой данных, поступающих от ядер-вычислителей, а при большем числе – перегружено главное ядро-сборщик, занятое обработкой поступающих данных от ядер-сборщиков.

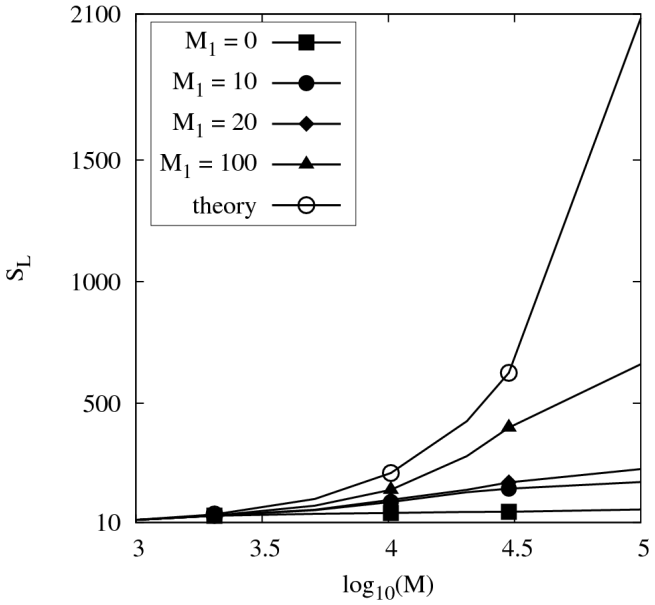

Рис. 5. Зависимость относительного ускорения S_L от общего числа моделируемых ядер M при разном числе ядер-сборщиков M_1 (горизонтальная ось – в логарифмическом масштабе)

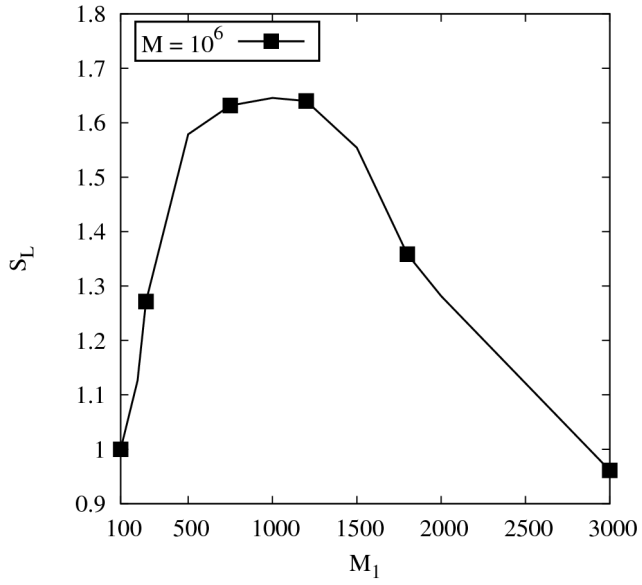

Рис. 6. Зависимость относительного ускорения S_L от числа ядер-сборщиков M_1 при общем числе моделируемых ядер M=10^6 .

6. Численное моделирование 3D сейсмических полей

Аналогичные исследования были проведены и для другого класса алгоритмов, основанного на применении разностного метода. В работе рассмотрен алгоритм численного моделирования 3D сейсмических полей в изотропной неоднородной упругой среде [13]. Такого вида задачи характеризуются большим объемом вычислений, поскольку область моделирования представляется достаточно подробно для проведения 3D моделирования.

Предполагается, что вычислительные модули кластера состоят из нескольких CPU и GPU. Поэтому 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. Способ декомпозиции расчетной области для организации параллельных вычислений представлен на рис 7.

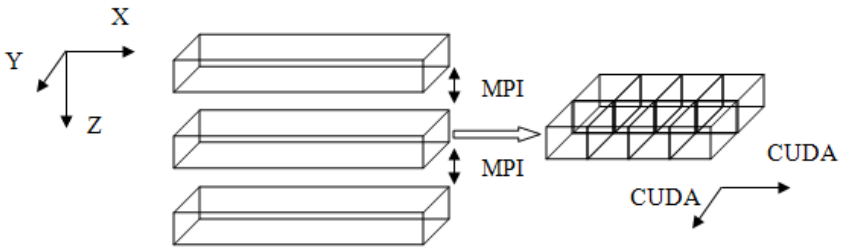

Рис. 7. Схема декомпозиции расчетной области.

7. Результаты имитационного моделирования сеточного метода

Для исследования реализации алгоритма численного моделирования на предполагаемую модель экзафлопсного компьютера выбран следующий критерий масштабируемости – 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. Исследование проводилось с использованием имитационного моделирования работы экзафлопсного суперкомпьютера в вышеупомянутой системе AGNES. Предполагается гибридная архитектура суперЭВМ. Для проверки адекватности результатов имитационного моделирования проводилось их сравнение с результатами работы данного алгоритма на гибридном кластере НКС-30Т+GPU ССКЦ.

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 8. 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%).

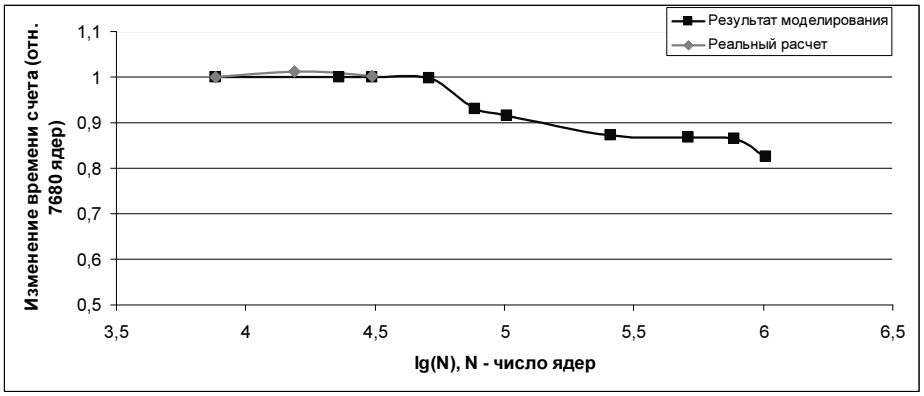

Рис. 8. Изменение времени расчета алгоритма численного моделирования в зависимости от числа вычислительных ядер (горизонтальная ось – в логарифмическом масштабе).

В заключение данного раздела приведем оценки необходимого количества ядер для реальных расчетов и соответственно для имитационной модели. В реальном расчете понадобилось 30720 ядер, а для имитационной модели только 12 вычислительных ядер! А при моделировании работы данного алгоритма с использованием до 1,5 млн. ядер в системе AGNES нам понадобилось только 144 ядра!

8. Заключение

Проведенное имитационное моделирование показало 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.

Следует отметить, что эксперименты показали нецелесообразность, а в некоторых случаях и невозможность использования соответствия «агент-ядро», что связано с естественными ограничениями, вызванными затратами на управление взаимодействием агентов в системе JADE. Однако вполне возможно моделирование одним агентом поведения множества вычислительных ядер. В этом случае моделирование, по крайней мере, для рассматриваемых алгоритмов, осуществляется достаточно эффективно без потери точности. Максимальное количество моделируемых ядер достигало 10^6 , что даёт обоснованную надежду на возможность практического применения системы AGNES для моделирования исполнения на супер ЭВМ параллельных программ и других классов.

Таким образом, в настоящее время для оценки масштабируемости вычислительного алгоритма на гибридном кластере можно рекомендовать следующее: составить схему выполнения программы; прогнать параллельную программу на небольшом количестве ядер от сотен до нескольких тысяч ядер; ввести в имитационную модель задержки, отображающие время счета на вычислительных ядрах, полученные из реальных расчетов; протестировать правильность расчета на имитационной модели путем сравнения на начальном участке реального и модельного расчетов; исследовать поведение алгоритма при большом количестве ядер (от сотен тысяч до миллионов вычислительных ядер); провести, при необходимости, коррекцию вычислительной схемы, реализующей данный алгоритм.

Система AGNES установлена в ЦКП ССКЦ ИВМиМГ СО РАН и доступна по ссылке http://www2.sscc.ru/PPP/Mat-Libr/agnes.htm.

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. F. L. Bellifemine, G. Caire, D. Greenwood, Developing Multi-Agent Systems with JADE // Wiley, 2007.
5. D. Podkorytov, A. Rodionov, H. Choo, Agent-based simulation system AGNES (AGent NEtwork Simulator) for networks modeling // Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, ICUIMC'12, 2012.
6. Д.И. Подкорытов. Агентно-ориентированная среда моделирования сетевых систем AGNES // Ползуновский вестник, 2012. № 2/1, с. 93-106.
7. Эксафлопсные суперЭВМ, контуры архитектуры. Труды конференции PACO-2012 [Электронный ресурс]. - Режим доступа: http://paco2012.ipu.ru/procdngs/B101.pdf.
8. М.А. Марченко, Г.А. Михайлов. Распределенные вычисления по методу Монте-Карло // Автоматика и телемеханика. 2007. Вып. 5. С. 157-170.
9. M.A. Marchenko, PARMONC - A Software Library for Massively Parallel Stochastic Simulation // LNCS. 2011. V. 6873. P. 302-315.

10. Страница библиотеки PARMONC на сайте ССКЦ КП СО РАН [Электронный ресурс]. - Режим доступа: http://www2.sscc.ru/SORAN-INTEL/paper/2011/parmonc.htm.

11. Boris Glinsky, Alexei Rodionov, Mikhail Marchenko, Dmitry Podkorytov, Dmitry Weins. Scaling the Distributed Stochastic Simulation to Exaflop Supercomputers // Proceedings of 2012 IEEE 14th International Conference on High Performance Computing and Communications , p. 1131-1136.

12. 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.

13. Б.М. Глинский, Д.А. Караваев, В.В. Ковалевский, В.Н. Мартынов 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.