

На правах рукописи

Востокин Сергей Владимирович

ГРАФИЧЕСКАЯ ОБЪЕКТНАЯ МОДЕЛЬ ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОВ И
ЕЕ ПРИМЕНЕНИЕ В ПРОГРАММНЫХ КОМПЛЕКСАХ ЧИСЛЕННОГО
МОДЕЛИРОВАНИЯ

Специальность: 05.13.18 — Математическое моделирование, численные
методы и комплексы программ

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
доктора технических наук

Самара 2007

Работа выполнена в Государственном образовательном учреждении высшего профессионального образования «Самарский государственный аэрокосмический университет имени академика С.П. Королева» (СГАУ) на кафедре информационных систем и технологий.

Научный консультант: доктор технических наук, профессор
Прохоров Сергей Антонович

Официальные оппоненты: доктор технических наук, профессор
Гергель Виктор Павлович

доктор технических наук, профессор
Олейников Александр Яковлевич

доктор технических наук, профессор
Кораблин Михаил Александрович

Ведущая организация: Федеральное государственное унитарное
предприятие Государственный научно-
производственный ракетно-космический центр
«ЦСКБ-Прогресс», г. Самара

Защита состоится 19 октября 2007 г. в 10 часов на заседании диссертационного совета Д 212.215.05 при Государственном образовательном учреждении высшего профессионального образования «Самарский государственный аэрокосмический университет имени академика С.П. Королева» по адресу: 443086, г. Самара, Московское шоссе, 34, ауд. 209.

С диссертацией можно ознакомиться в библиотеке СГАУ.

Автореферат разослан «___» _____ 2007 г.

Ученый секретарь
диссертационного совета
д.т.н., профессор

Калентьев А.А.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность проблемы. Прогресс вычислительной техники оказывает значительное влияние на развитие методов математического моделирования. Производительность суперкомпьютеров за последнее десятилетие возросла примерно в тысячу раз. Сегодня для проведения единичного численного эксперимента доступна очень большая вычислительная мощность, измеряемая десятками терафлоп. Рост производительности компьютеров позволяет строить качественно новые модели сложных физических, химических, биологических, экономических, технических, социальных и других систем. С помощью данных моделей удастся наблюдать процессы и предсказывать явления ранее недоступные в силу низкой производительности компьютеров. Это расширяет область применения численных моделей и позволяет использовать более универсальные подходы к моделированию.

Новые возможности численного моделирования связаны с интенсивным развитием сетевых технологий. Агрегация вычислительных ресурсов позволяет добиться очень высокой пиковой производительности и снижения стоимости вычислительных экспериментов. С другой стороны, имеется ряд важных задач математического моделирования, которые обуславливают потребность во все более мощной вычислительной технике. Это задачи параметрической оптимизации; задачи исследования асимптотического поведения моделей, основанных на системах дифференциальных уравнений; сеточные и конечноэлементные модели; модели взаимодействия частиц.

Однако применение агрегированных вычислительных ресурсов в численном моделировании сдерживается из-за отсутствия средств и методов организации вычислений, адекватных новому типу вычислительной среды. В агрегированной среде исполнения возникают проблемы организации вычислений, обусловленные программной и аппаратной гетерогенностью, ненадежностью и распределенным характером среды исполнения, большим числом процессов, сложностью управления ресурсами. Обычные методы реализации параллельных численных алгоритмов с использованием MPI сложно масштабируются на новый тип вычислительных сред. Технологии, реализованные в грид-системах на основе программного обеспечения Globus, gLite, ARC, решают задачу организации крупно-гранулярных вычислений. В альтернативных индустриальных технологиях, например, в системе координации Jini, технологиях RMI, CORBA, DCOM не предлагается способов описания вычислительных процессов в целом.

В связи с этим, *актуальной научно-технической проблемой является реализация численных методов на новом типе вычислительных систем, образованных за счет агрегации вычислительных ресурсов на основе метакомпьютерных и грид-технологий.*

Объект исследования. Эффективные методы отображения алгоритмов, записанных на процедурных языках, на архитектуру однопроцессорного компьютера используются около тридцати лет. Они позволяют при построении численного метода исключить из рассмотрения аппаратные особенности компьютера, такие как архитектура процессора. В случае параллельной

реализации численного метода перед исследователем возникают две сложные проблемы. Во-первых, требуется выявить параллельно исполняющиеся части алгоритма и построить корректную синхронизацию, согласующую их работу. Большое число возможных состояний вычислительного процесса ведет к тому, что построение правильной структуры алгоритма для параллельной реализации модели становится нетривиальной задачей. Во-вторых, в отличие от последовательной реализации численного метода, приходится учитывать воздействие аппаратуры на ход вычислений. Поведение оборудования влияет на логику работы алгоритма (возможны разные варианты выполнения вычислений в зависимости от случайных факторов); на эффективность (существенным является соотношение скорости вычислений и обменов); на возможность успешного завершения (в распределенной системе велика вероятность сбоев). Таким образом, совокупность параллельных алгоритмов численного моделирования и реализующего их оборудования представляет собой сложную дискретную систему, для описания которой необходимо разрабатывать специальные формальные методы. В связи с этим **объектом исследования** в работе является новая модель пространственно-распределенной дискретной системы, описывающая совместные свойства параллельного алгоритма численного моделирования и аппаратуры.

Традиционный подход к преодолению сложности разработки численных методов для параллельных компьютеров лежит в области автоматизации программирования. Ведутся исследования, цель которых создание распараллеливающих компиляторов, анализаторов последовательного кода на предмет выявления параллелизма. Разрабатываются непроцедурные методы программирования численных моделей. В данном направлении известны работы А.П. Ершова, В.В. Воеводина, В.А. Вальковского, Л.Дж. Баера (L.J. Baer) и многих других отечественных и зарубежных ученых. Такие подходы являются эффективными для узкого класса алгоритмов и вычислительных систем.

Представление алгоритма и среды его исполнения в форме модели дискретной системы широко применяется для анализа. В этой области работали такие ученые как А. Нуэли (A. Pnueli), К.М. Чанди (K.M. Chandy), Я. Мизра (J. Misra), Л. Лампорт (L. Lamport) и другие. Однако анализу подвергаются готовые реализации алгоритмов. Поэтому методы анализа обычно не учитывают вопрос о существовании эффективной реализации той или иной модели, хотя известно, что некоторые простые дискретные модели не возможно эффективно реализовать в распределенных средах.

В диссертации рассматривается альтернативный подход, использующий специальные модели в качестве основы алгоритмизации и программирования численных методов. Эти модели представляют собой схемы параллельных алгоритмов численного моделирования. Данный подход был предложен М.И. Коулом (M.I. Cole) и развивается в работах С. Макдональда (S. MacDonald), П.К. Берзигиярова и других авторов. Тем не менее, недостаточно исследованными следует считать вопросы построения эффективно реализуемых универсальных моделей дискретных систем с использованием объектно-ориентированной парадигмы и методов

визуализации алгоритмов, а также вопросы изучения взаимосвязи между алгоритмом и реализующей вычислительной средой методами имитационного моделирования.

Практическая ценность данного подхода заключается в построении численных моделей в актуальных предметных областях, эффективно реализуемых на различных параллельных вычислительных системах. Форма описания таких моделей не связана с особенностями среды исполнения, а для их программирования используются стандартные средства. Однако практическая реализация предлагаемого подхода требует разработки новых математических методов описания статической и динамической структуры алгоритмов численного моделирования, а также комплекса программ синтеза и анализа соответствующих моделей.

Цель и задачи исследования. Целью работы является решение актуальной научно-технической проблемы реализации алгоритмов численного моделирования в произвольной среде исполнения, представляющей собой совокупность разнотипных вычислительных ресурсов, связанных общей коммуникационной средой.

В соответствии с поставленной целью определены *задачи диссертации*.

1) Проанализировать известные модели, методы и программные системы в области параллельного и распределенного программирования с целью определения требований к модели дискретной системы.

2) На основе сформулированных требований построить модель дискретной системы, которая описывает алгоритмы численного моделирования в произвольной среде исполнения.

3) Обосновать пригодность построенной модели для описания алгоритмов численного моделирования и возможность эффективного исполнения модели в распределенной вычислительной среде.

4) Разработать язык и метод описания прикладных программ численного моделирования в терминах предложенной модели дискретной системы.

5) Выполнить исследование адекватности предложенной модели дискретной системы в вычислительных экспериментах с использованием дискретно-событийной имитационной модели.

6) Разработать комплекс программных средств синтеза и анализа моделей, библиотеку программ численного моделирования.

Методы исследования. При проведении исследований были использованы элементы теории множеств, математической логики, теории графов. Результаты теоретических исследований подтверждены вычислительными экспериментами, экспериментами с имитационными моделями, а также реализацией прикладных алгоритмов численного моделирования в разных предметных областях.

Основные научные результаты, полученные автором и выносимые на защиту.

1) Модель дискретной системы для представления алгоритмов численного моделирования в произвольной среде исполнения.

2) Обоснование адекватности предложенного метода моделирования для описания численных моделей, реализуемых в распределенных вычислительных средах.

3) Методика построения параллельных алгоритмов численного моделирования, включающая средства описания схем вычислений, алгоритмы кодирования схем, примеры прикладных численных методов.

4) Результаты анализа эффективности предложенного метода моделирования.

5) Программный комплекс дискретного моделирования вычислительных процессов и библиотека прикладных программ численного моделирования.

Научная новизна результатов диссертации. Научная новизна диссертации заключается в том, что в ней впервые получены следующие результаты.

1) Построена новая объектная модель GraphPlus, которая, в отличие от известных моделей, пригодна для описания вычислительных процессов, возникающих при реализации алгоритмов численного моделирования на широком классе вычислительного оборудования (сети из персональных ЭВМ, многопроцессорные ЭВМ с общей памятью, кластерные системы и специализированные суперЭВМ, а также системы, построенные путем агрегации этих ресурсов).

2) Описан метод визуализации модели, на котором основывается декомпозиция параллельного вычислительного алгоритма на совокупность последовательных процедур, а также метод построения алгоритмов, реализующих функциональные отношения модели по ее визуальному представлению в виде множества диаграмм. Данный метод в отличие от известных подходов (CODE, HeNCE), соответствует современной практике программирования.

3) На основе объектной модели предложен новый метод построения и формального описания каркасных библиотек численного моделирования, который, в отличие от известных методов, рассчитан на произвольную среду исполнения и строго формализован.

4) Предложена общая структура алгоритмов управления вычислениями с использованием модельного описания в форме паттерна, названного «Постоялец-Посетитель». Ее применение, в отличие от известных подходов, позволяет отделить код численного метода от управляющего кода при сохранении эффективности вычислений; она пригодна для широкого класса вычислительных сред.

5) Предложен метод имитационного моделирования вычислительных процессов, описанных в терминах графической объектной модели, и получены новые экспериментальные результаты дискретно-событийного моделирования исполнения вычислений в гетерогенных вычислительных средах с флуктуирующими характеристиками производительности и пропускной способности.

Обоснованность и достоверность результатов диссертации. Предложенные в диссертации новые решения строго аргументированы и критически оценены по сравнению с другими известными результатами.

Построенные в работе модели процессов базируются на использовании апробированных формальных методов спецификации дискретных систем. Достоверность выводов подтверждена результатами проведенных вычислительных экспериментов на реальном оборудовании и имитационными экспериментами с моделями.

Практическая значимость результатов диссертации. Практическое использование полученных результатов позволяет:

1) повысить надежность программного обеспечения численного моделирования за счет применения формальных моделей при описании распределенных и параллельных алгоритмов, а также за счет наглядной формы их представления в виде ориентированных размеченных графов специального вида;

2) преодолеть сложности, связанные с необходимостью при реализации численных моделей включать в них код, поддерживающий специфические для гетерогенных и ненадежных сред методы управления вычислениями, в форме алгоритмов балансировки загруженности и алгоритмов обеспечения отказоустойчивости;

3) проводить ресурсоемкие вычислительные эксперименты на неспециализированном оборудовании, таком как рабочие станции, связанные сетью, что с одной стороны понижает материальные затраты на организацию вычислительных экспериментов, а с другой стороны позволяет интегрировать большие вычислительные мощности по сравнению с одиночными кластерными системами;

4) упростить подготовку модельных экспериментов с привлечением больших вычислительных ресурсов на основе применения каркасных библиотек, описывающих типовые вычислительные процессы.

Практическим результатом диссертации является разработка программного комплекса параллельных и распределенных вычислений GraphPlus, включающего утилиты описания визуального представления схем вычислительных процессов, транслятор схем процессов, кодогенератор, интерпретатор, различные варианты среды времени исполнения, в том числе для проведения имитационных экспериментов с моделями процессов и другие. Основные компоненты программного комплекса официально зарегистрированы Федеральной службой по интеллектуальной собственности, патентам и товарным знакам: «Модуль многопоточного исполнения моделей программного комплекса проекта Граф Плюс» — свидетельство об официальной регистрации №2007611206; «Транслятор моделей программного комплекса проекта Граф Плюс» — свидетельство об официальной регистрации №2007611201. Общий объем исходного кода данных компонент превышает 8000 строк.

Результаты диссертации внедрены в ГОУ ВПО «Самарский государственный аэрокосмический университет», ГОУ ВПО «Самарский государственный технический университет», ГОУ ВПО «Поволжская государственная академия телекоммуникаций и информатики», ГОУ ВПО «Самарский государственный экономический университет», ОАО «Самаранефтегаз» (в разработках ООО ИК «СИБИНТЕК»), ООО «Mercury

Development Russia», ГУЗ «Самарский областной медицинский информационно-аналитический центр». Программный комплекс дискретного моделирования вычислительных процессов, примеры алгоритмов и технические описания доступны на официальном сайте Самарского государственного аэрокосмического университета по адресу <http://graphplus.ssau.ru>.

Представленные в работе методы моделирования вычислений в совокупности со специальной формой представления библиотек, приемами декомпозиции прикладного и системного кода и разделением ролей в группе исследователей, организующей модельные эксперименты, могут быть использованы для разработки метакомпьютерных и грид-приложений.

Связь исследований с научными проектами. Теоретические и практические результаты диссертационной работы использованы при выполнении госбюджетных работ в рамках персонального гранта Министерства образования Российской Федерации и Правительства Самарской области на проведение исследований в области гуманитарных, общественных, технических наук и естествознания по теме «Разработка методов визуального проектирования управляющих алгоритмов для систем распределенной пакетной обработки заданий», шифр темы 17Г-Р077-090-050 В2; при выполнении исследовательских работ по гранту Американского фонда гражданских исследований и развития (CRDF Project SA -014-02).

Апробация работы. Результаты работы докладывались и обсуждались на всероссийских и международных научно-технических конференциях:

- Научно-технической конференции с международным участием «Перспективные информационные технологии в научных исследованиях, проектировании и обучении», Самара, 2006 г.;

- Третьей Всероссийской научной конференции «Математическое моделирование и краевые задачи», Самара, 2006 г.;

- Второй международной конференции «Распределённые вычисления и грид-технологии в науке и образовании», Дубна, 2006 г.;

- Третьей международной конференции «Параллельные вычисления и задачи управления», Москва, 2006 г.

- Международных конференциях «Надежность и качество», Пенза, 1999, 2002-2006 гг.;

- Всероссийской межвузовской научно-практической конференции «Компьютерные технологии в науке, практике и образовании», Самара, 2005 г.;

- Четвертом международном научно-практическом семинаре «Высокопроизводительные параллельные вычисления на кластерных системах», Самара, 2004 г.;

- Международной конференции «Интерактивные системы. Проблемы человеко-машинного взаимодействия», Ульяновск, 2003 г.;

а также на Всероссийской молодежной научной конференции «XXIII Гагаринские чтения», Москва, 1997; Всероссийской студенческой научной конференции «Королевские чтения», Самара, 1997; Первой и Второй международной молодежной школе-семинаре БИКАМП, Санкт-Петербург, 1998 и 1999; IV Всероссийской научной конференции студентов и аспирантов

«Техническая кибернетика, радиоэлектроника и системы управления», Таганрог, 1998; VI Международной конференции “Математика, компьютер, образование”, Москва, Пущино, 1999.

Публикации. По теме диссертации опубликовано 42 работы, в том числе 3 монографии, 29 статей, из них 13 — в изданиях, входящих в перечень ведущих рецензируемых научных журналов и изданий, выпускаемых в Российской Федерации, в которых должны быть опубликованы основные научные результаты диссертации на соискание ученой степени доктора наук; а также получено 2 свидетельства о регистрации программ для ЭВМ, опубликован сайт в сети Интернет на сервере Самарского государственного аэрокосмического университета.

Структура и объем работы. Диссертация состоит из введения, пяти глав, заключения, библиографического списка и приложений; содержит 266 страниц основного текста, 84 рисунка, 18 таблиц. Библиографический список включает 206 наименований литературы.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность работы, дана ее общая характеристика, сформулированы цель и задачи исследования.

В первой главе приведен обзор методов, моделей и программных систем в области параллельного и распределенного программирования. Современные аппаратные технологии позволяют использовать как специализированные системы, так и компьютеры, связанные сетью, для организации параллельных вычислений. Возможность использования большого числа недорогих вычислительных ресурсов делает актуальным поиск новых методов построения алгоритмов численного моделирования, приспособленных для исполнения в произвольной вычислительной среде. В связи с этим в главе 1 приведен анализ известных формальных методов, на основании которого формулируются требования к предлагаемой в диссертации модели вычислений.

Модель вычислений должна иметь четкую и однозначную интерпретацию. Поэтому представляет интерес способ описания семантики исполнения модели. Обычно в моделях предполагается наблюдаемость глобального состояния. Примером являются модели Р. Милнера (R. Milner), Ч. Хоара (C.A.R. Hoare), темпоральная логика А. Нуэли (A. Pnueli), сеть Петри. И. Халперн (J. Halpern) и И. Мозес (Y. Moses) предложили модель, в которой глобальное состояние неизвестно, а вычисления интерпретируются как накопление сведений при взаимодействии. Все эти модели основаны на принципе локальности действий и представлении вычислений в форме истории. Историю представляют как последовательность или дерево событий (метод Р. Милнера (R. Milner)) либо состояний. Использование состояний или смешанная форма представления историй характерна для логических моделей, например, в логике Л. Лампорта (L. Lamport). Существует несколько способов применения формализмов. Выделяются методы, целью которых является синтез параллельных программ с желаемыми свойствами. Методы распараллеливания используются в процедурных описаниях (проект V-Ray), в функциональных и логических

языках (проект Т-системы) и непроцедурных предметно-ориентированных описаниях (язык Норма). При распараллеливании процедур различают задачи о распараллеливании циклических, ациклических фрагментов и распараллеливании выражений. Исследовались методы статического и динамического распараллеливания. Специальным методом синтеза является предложенный К.М. Чанди (K.M. Chandy) и И. Мизра (J. Misra) метод последовательного улучшения, позволяющий постепенно переходить от спецификации к реализации параллельного алгоритма. Наиболее популярны методы анализа, обосновывающие свойства имеющихся объектов. Большинство методов используют подход, предложенный С. Овицки (S. Owicki) и Д. Грисом (D. Gries). Формальные теории имеют форму алгебр или логик. Исследование моделей может выполняться путем «исчерпывающего» анализа историй, обобщенных рассуждений и имитационными методами.

Новая модель вычислений должна учитывать особенности известных моделей. В параллельном программировании выделяют модели, не использующие понятие потока исполнения или асинхронные (Б.Б. Нестеренко, В.А. Марчук). Это модели с событийным, потоковым и динамическим управлением. Более традиционными и удобными в реализации являются модели с потоком управления. При реализации также имеет значение способ структурирования кода, который может соответствовать модели SPMD «одна программа - много данных» либо MPMD «много программ - много данных».

Для описания распределенных вычислений наиболее адекватны модели процессов. Необходимыми допущениями в таких моделях являются рассмотрение отказов и учет времени доставки сообщений. Н. Линч (N. Lynch) и Л. Лампорт (L. Lamport) классифицируют модели процессов на три группы: модели с разделяемыми переменными, модели распределенных процессов и другие модели, к числу которых относят модель Ч. Хоара (C.A.R. Hoare), рандеву, удаленный вызов процедуры (RPC). Группу моделей распределенных процессов разделяют по признакам топологии коммуникаций, синхронности, отказам и способам буферизации. Несмотря на отсутствие единой теории, объединяющей модели, описаны алгоритмы, обосновывающие возможность организации вычислений в распределенных ненадежных средах, например, алгоритм распределенной транзакции М.Д. Скина (M.D. Skeen); методы синхронизации часов Д. Долева (D. Dolev), И. Халперна (J. Halpern), Х.Р. Стронга (H.R. Strong); алгоритм получения мгновенного снимка К.М. Чанди (K.M. Chandy) и Л. Лампорта (L. Lamport); протоколы линка систематизированные А.В. Ахо (A.V. Aho), И. Ульманом (J. Ullman) и другие. Поэтому необходимо, чтобы специальная модель распределенных вычислений очевидным образом сводилась к известным моделям, для которых получены эффективные алгоритмы реализации.

При разработке новой модели вычислительных процессов требуется учитывать влияние системного программного обеспечения, а также средств автоматизации программирования. От этого зависит удобство использования и совместимость нового метода с другими аналогичными технологиями. В качестве общепринятых средств программирования в главе 1 рассмотрены API

операционных систем (Win32, POSIX); стандартные средства Open MP, MPI, PVM; сокет; механизмы удаленного вызова; промежуточное программное обеспечение грид-систем (GT, gLite, ARC и др.). Также рассмотрены языковые средства; прикладные библиотеки на основе MPI и PVM; библиотеки и системы программирования, использующие каркасы; инструменты автоматизированного распараллеливания; средства автоматизированного проектирования.

В результате анализа известных средств и методов параллельных и распределенных вычислений сформулированы требования к модели описания вычислений в гетерогенной распределенной среде исполнения. Во-первых, для достижения максимальной эффективности в модели должны быть средства явного описания параллелизма и декомпозиции данных, а виртуальная машина, исполняющая модель, имитировать надежный многопроцессорный компьютер с общей памятью. Во-вторых, модель должна эффективно реализовываться современными языковыми и системными средствами. В-третьих, модель должна обеспечивать построение повторно используемых библиотек численного моделирования.

Во второй главе формализована спецификация модели вычислений GraphPlus, изучена ее адекватность при описании вычислительных процессов и рассмотрен способ наглядного графического представления.

Предлагаемая модель основана на объектной декомпозиции параллельного алгоритма, реализующего численный метод. Все используемые в алгоритме переменные сгруппированы в объекты. Структура произвольного параллельного алгоритма образуется объектами двух типов — «неподвижными» или Р-объектами и «подвижными» или М-объектами.

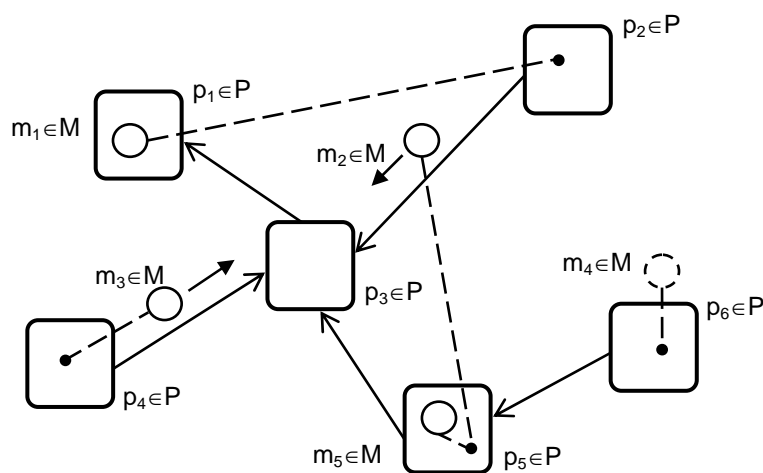


Рис. 1. Мгновенный снимок структуры параллельного вычислительного процесса в модели GraphPlus

Обозначим символом P множество всех Р-объектов, а M — множество всех М-объектов. Для изображения Р-объектов на рис. 1 используются квадраты (объекты $p_1 \dots p_6$ из множества P). Для изображения М-объектов используются кружки (объекты $m_1 \dots m_5$ из множества M). При помощи сплошных и пунктирных линий на рис. 1 показаны статические (не меняющиеся в процессе исполнения) отношения между объектами. Пунктирной линией показаны связи между Р- и М- объектами. Р-объект может

быть связан с несколькими М-объектами: $L : P \times M \rightarrow \{0,1\}$. При этом М-объект связан строго с одним Р-объектом: $\forall x \in M : \overline{\exists p_1, p_2 : p_1 \neq p_2 \wedge L(p_1, x) = 1 \wedge L(p_2, x) = 1}$. Отношение L задает подмножество «локальных» М-объектов для каждого Р-объекта, которое может быть пустым. Но каждый М-объект обязательно связан с Р-объектом: $\forall x \in M, \exists y \in P : L(y, x) = 1$. Связи между Р-объектами показаны сплошными ориентированными линиями. Соответствующее данной связи отношение обозначается как $N : P \times P \rightarrow \{0,1\}$. Подграф, описывающий отношение N , является ориентированным, то есть для произвольной модели $N(x, y) = 1 \not\Rightarrow N(y, x) = 1$.

Вычисления на модели интерпретируются как одновременный обход несколькими М-объектами структуры из Р-объектов. Часть М-объектов в некоторый момент времени может находиться в неактивном состоянии. На рис. 1 это объект m_4 , обозначенный пунктирным контуром. Находясь в активном состоянии, М-объекты могут либо перемещаться по связям между Р-объектами (показано стрелками), либо посещать Р-объект (передается помещением кружка внутрь квадрата). В момент посещения происходит изменение состояния взаимодействующей пары объектов, и возможна активация неактивных М-объектов, связанных с посещаемым Р-объектом. После посещения объекта p М-объект может направиться к любому из соседних объектов $p' : N(p, p') = 1$, снова посетить объект p или перейти в неактивное состояние.

Далее в главе 2 на основе рассмотренного вербального описания строится формальная спецификация модели. Она использует математический аппарат TLA (temporal logic of actions), предложенный Лампортом (L. Lamport)¹⁾. Введем функции, сопоставляющие объект модели GraphPlus переменной: $o[] : Obj \rightarrow Var$. Будем считать, что все темпоральные переменные в формулах модели вычислений заданы через функции o , а темпоральные константы заданы обычным способом. Определим множества объектов модели GraphPlus:

$$\begin{aligned} P &\subset Obj \quad M \subset Obj \quad \varepsilon \in Obj; \\ P \cap M &= \emptyset, \quad P \cap \{\varepsilon\} = \emptyset, \quad M \cap \{\varepsilon\} = \emptyset; \\ P_+ &= P \cup \{\varepsilon\}, \quad M_+ = M \cup \{\varepsilon\}. \end{aligned}$$

Здесь ε описывает специальный «нуль» объект, а P_+ и M_+ — множества Р- и М-объектов, расширенные «нуль» объектом. Обозначим домены значений объектов модели, как $D_P \subset Val$ и $D_M \subset Val$. Каждому М-объекту будут соответствовать переменные модели, задаваемые функциями $\tilde{m}[] : M \rightarrow V_{\tilde{M}}$ $m[] : M \rightarrow V_M$, а каждому Р-объекту — функциями $\tilde{p}[] : P \rightarrow V_{\tilde{P}}$ $p[] : P \rightarrow V_P$ $\tilde{n}[] : P \rightarrow V_N$.

Переменные М-объектов $\tilde{m}[]$ принимают значения Р-объектов, которые они собираются посетить. Переменные Р-объектов $\tilde{p}[]$, в свою очередь, принимают значения М-объектов, посещающих их в некоторый момент времени. Переменные $\tilde{n}[]$ служат для хранения Р-объектов, к которым

¹⁾ Lamport L. Specifying Systems. The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley, 2002.

произойдет переход после «посещения». Наконец, переменные $m[]$ и $p[]$ определяют состояние конкретного моделируемого вычислительного процесса. Множества переменных различны и попарно не пересекаются:

$$\begin{aligned} V_{\bar{M}} \subset \text{Var}, \quad V_M \subset \text{Var}, \quad V_P \subset \text{Var}, \quad V_{\bar{P}} \subset \text{Var}, \quad V_N \subset \text{Var}; \\ V_{\bar{M}} \cap (V_M \cup V_P \cup V_{\bar{P}} \cup V_N) = \emptyset; \quad V_M \cap (V_{\bar{M}} \cup V_P \cup V_{\bar{P}} \cup V_N) = \emptyset; \\ V_P \cap (V_M \cup V_{\bar{M}} \cup V_{\bar{P}} \cup V_N) = \emptyset; \quad V_{\bar{P}} \cap (V_M \cup V_{\bar{M}} \cup V_P \cup V_N) = \emptyset; \\ V_N \cap (V_M \cup V_{\bar{M}} \cup V_P \cup V_{\bar{P}}) = \emptyset. \end{aligned}$$

Отношения инцидентности N и L между объектами модели будем записывать в префиксной форме в виде специальных предикатов $N(\cdot, \cdot)$ и $L(\cdot, \cdot)$.

Для описания динамики изменения состояния в модели GraphPlus определим четыре функции:

$$\begin{aligned} newP(\cdot, \cdot) : D_P \times D_M \rightarrow D_P & \text{— новое состояние посещаемого P-объекта;} \\ newM(\cdot, \cdot) : D_P \times D_M \rightarrow D_M & \text{— новое состояние посещающего M-объекта;} \\ activate(\cdot, \cdot) : M \times D_P \times D_M \rightarrow P_+ & \text{— функция активации локальных M-объектов;} \\ next(\cdot, \cdot) : D_P \times D_M \rightarrow P_+ & \text{— переход M-объекта к следующему P-объекту.} \end{aligned}$$

Обозначив множество всех отношений в модели через $\mathfrak{R} = N \cup L \cup newP \cup newM \cup activate \cup next$, определим модель GraphPlus как

$$\Psi = (P, M, \mathfrak{R}, \Phi), \quad (1)$$

где Φ — формула TLA, описывающая динамику поведения вычислительного процесса. Структура формулы определяется выражениями (2)-(9) в соответствии с графом модели, пример которого показан на рис. 1. Формула строится с учетом обеспечения минимальной подробности, необходимой для адекватной передачи состояния P- и M- объектов; минимального влияния на реализацию модели; возможности изучения на модели состояний, отображающих одновременность выполнения переходов и посещений для M-объектов.

Формула, описывающая дискретную систему в логике TLA по Лампорту,

$$\Phi \stackrel{def}{=} I \wedge \square [N]_f \wedge F. \quad (2)$$

Предикат начального состояния вычислений в модели GraphPlus

$$\begin{aligned} I \stackrel{def}{=} \wedge \exists y \in M : (\tilde{m}[y] \neq \varepsilon \wedge \tilde{m}[y] \in P) \wedge \forall x \in P : \tilde{p}[x] = \tilde{n}[x] = \varepsilon \\ \wedge \forall x \in P, \forall y \in M : p[x] \in D_P \wedge m[y] \in D_M. \end{aligned} \quad (3)$$

Начальное состояние системы (3) предусматривает наличие хотя бы одного M-объекта в состоянии перемещения и правильно проинициализированных переменных численного метода. Функция состояния, представляющая собой кортеж из всех переменных, используемых для представления конкретной модели,

$$f \stackrel{def}{=} (\tilde{p}[x] : x \in P; \tilde{m}[y] : y \in M; \tilde{n}[x] : x \in P; p[x] : x \in P; m[y] : y \in M). \quad (4)$$

Формула для N описывает множество допустимых действий (событий) в модели вычислений. Это множество разбито на два подмножества A_1 и A_2 . Их атомарность (взаимная независимость) следует из определений (7)-(9). Формула для N имеет вид

$$N \stackrel{def}{=} \begin{aligned} &\vee \exists x \in P, \exists y \in M : A_1(x, y) \\ &\vee \exists x \in P, \exists y \in M : A_2(x, y) \end{aligned} \quad (5)$$

Формула, описывающая условия «справедливости» выполнения действий для модели, записывается в виде

$$F \stackrel{def}{=} \begin{aligned} &\wedge \forall x \in P, \forall y \in M : SF_f(A_1(x, y)) \\ &\wedge \forall x \in P, \forall y \in M : WF_f(A_2(x, y)) \end{aligned} \quad (6)$$

Группа действий A_1 задается формулами (7), (8).

$$A_1(x, y) \stackrel{def}{=} \begin{aligned} &\wedge \tilde{p}[x] = \varepsilon \wedge \tilde{m}[y] = x \\ &\wedge \tilde{p}'[x] = y \\ &\wedge p'[x] = newP(p[x], m[y]) \\ &\wedge m'[y] = newM(p[x], m[y]) \\ &\wedge \tilde{n}'[x] = next(p[x], m[y]) \\ &\wedge Local(x, y) \\ &\wedge unchanged(\bigvee_{\Phi} \setminus (\{\tilde{p}[x], p[x], \tilde{n}[x], m[y]\} \cup \{\tilde{m}[k] : L(x, k)\})) \end{aligned} \quad (7)$$

$$Local(x, y) \stackrel{def}{=} \begin{aligned} &\forall k \in M, L(x, k) : \\ &\vee \wedge \tilde{m}[k] = \varepsilon \\ &\wedge \tilde{m}'[k] = activate(k, x, y) \\ &\vee \wedge \tilde{m}[k] \neq \varepsilon \\ &\wedge \tilde{m}'[k] = \tilde{m}[k] \end{aligned} \quad (8)$$

В формулах (7), (8) переменные x и y являются темпоральными константами. Каждой паре из P -объекта и M -объекта сопоставляется индивидуальное действие A_1 . Формула (7) утверждает, что действие A_1 возможно, если P -объект не занят ($\tilde{p}[x] = \varepsilon$), а M -объект из рассматриваемой пары (x, y) направляется к P -объекту x ($\tilde{m}[y] = x$). Результатом действия A_1 являются следующие изменения в состоянии модели. Пара объектов (x, y) переходит в состояние посещения ($\tilde{p}'[x] = y$). Изменяется состояние P -объекта, описывающего моделируемый вычислительный процесс ($p'[x] = newP(p[x], m[y])$). Изменяется состояние посещающего M -объекта

($m'[x] = newM(p[x], m[y])$). Вычисляется следующий Р-объект для посещения, и он сохраняется в специальной переменной ($\tilde{n}'[x] = next(p[x], m[y])$). Происходит активация некоторых неактивных М-объектов, связанных с рассматриваемым Р-объектом ($\tilde{m}'[k] = activate(k, x, y)$).

Для соблюдения атомарности в формулах (7) и (9) учитывается, что остальные переменные модели не изменяются. Для этого служит специальное действие *unchanged*. В нем V_Φ — все переменные, входящие в формулу Φ (2).

Действие A_2 описывает, каким образом изменяется состояние Р-объекта после посещения (9).

$$\begin{aligned}
 A_2(x, y) &\stackrel{def}{=} \wedge \tilde{p}[x] = y \wedge \tilde{m}[y] = x \\
 &\wedge \tilde{p}'[x] = \varepsilon \\
 &\wedge \tilde{m}'[y] = \tilde{n}[x] \\
 &\wedge N(\tilde{m}[y], \tilde{m}'[y]) \\
 &\wedge unchanged(V_\Phi \setminus \{\tilde{p}[x], \tilde{m}[y]\})
 \end{aligned} \tag{9}$$

Условием готовности действия A_2 является наступление состояния посещения ($\tilde{p}[x] = y \wedge \tilde{m}[y] = x$). Действие состоит в присвоении переменной М-объекта $\tilde{m}[y]$ значения следующего посещаемого Р-объекта или нуля-объекта в случае перехода в неактивное состояние. Направление перехода предварительно сохраняется в переменной $\tilde{n}[x]$ в действии A_1 . Признаком перехода Р-объекта в свободное состояние является смена значения переменной $\tilde{p}[x]$ на ε . Другие переменные, кроме переменных $\tilde{p}[x]$ и $\tilde{m}[y]$, при выполнении действия A_2 не изменяются.

Множество допустимых действий связано с графической структурой модели. Действия (7) соотносят каждый подвижный М-объект каждому неподвижному Р-объекту, определяя, при каких условиях возможно соответствующее посещение и последующая активация новых М-объектов. Они также описывают, как изменяется состояние взаимодействующей группы объектов. Действие (7) должно быть «справедливо в строгом смысле» для предотвращения ситуации отталкивания одного М-объекта другим во время посещения некоторого Р-объекта. При реализации модели на многопоточной машине это обеспечивается путем организации очереди запросов на посещение с операцией помещения в очередь, реализованной в критической секции.

Действия (9) задают переход Р-объекта в исходное состояние после посещения, а также управляют переходом М-объекта к следующему Р-объекту или в неактивное состояние после посещения Р-объекта. Для действий (9) достаточна «справедливость в слабом смысле», так как выполнение других действий не может отменить условий их готовности. При реализации модели это гарантируется диспетчером потоков операционной системы.

Далее для исследования адекватности приводится описание простейших моделей процессов, описывающих взаимодействие конкурентного и кооперативного типа. Для этого выполняется табличное определение функций вычислительной модели *activate*, *next*, *newP*, *newM*, отношений L , N и предиката

начального состояния. В предположении о временах выполнения действий модели строятся истории, для которых формулы вида (2) интерпретируются в истину и выполняются соответствующие утверждения о возможности эффективной реализации модели в распределенной среде исполнения.

Для практического использования модели предложен способ графического представления функций $activate$, $next$, $newP$, $newM$ и отношений инцидентности L , N . Идея заключается в изображении каждого Р-объекта в виде ориентированного размеченного графа. Граф описывает алгоритм вычисления функций $activate$, $next$, $newP$, $newM$, а также представляет локальные М-объекты и связи с остальными Р-объектами.

Определим для графа Р-объекта множество пометок вершин. Данное множество будет включать в себя четыре попарно не пересекающихся класса пометок: $In \subseteq \{in\} \times F$, $Out \subseteq \{out\} \times F \times P$, $Do \subseteq \{do\} \times F$ и $Count \subseteq \{count\} \times F \times N$, где N — множество натуральных чисел. Пометки из данных классов на графе будем записывать в виде $in(\cdot)$, $out(\cdot, \cdot)$, $do(\cdot)$ и $count(\cdot, \cdot)$. Элементами множества F в определении классов пометок вершин являются функции, изменяющие состояние пары из М- и Р-объекта: $F: D_P \times D_M \rightarrow D_P \times D_M$. Пусть для каждого графа, обозначающего Р-объект, определен способ нумерации вершин. Тогда функцию пометок вершин для графа можно записать в виде $M(V): N \rightarrow In \cup Out \cup Do \cup Count$.

Аналогичным образом введем функцию пометок дуг графа $M(E): N \times N \rightarrow Ift \cup If \cup New$. Множество пометок дуг включает три класса: $Ift \subseteq \{ift\} \times M \times N$, $If \subseteq P \times N$ и $New \subseteq M \times N$, где P обозначает предикат на паре состояний М- и Р-объектов $P: D_P \times D_M \rightarrow \{0,1\}$. Пометки из данных классов на графе будем записывать в виде $ift(\cdot, \cdot)$, $if(\cdot, \cdot)$, $new(\cdot, \cdot)$.

Примером использования пометок служат графы кооперативного взаимодействия процессов, показанные на рис. 2. Здесь использованы специальные обозначения для функции, не изменяющей состояние М- и Р-объекта из пары (=), и предиката, возвращающего истинное значение для произвольных значений состояний М- и Р-объекта из пары (I). Для наглядности вершины и дуги обозначены способом, соответствующим классу пометок.

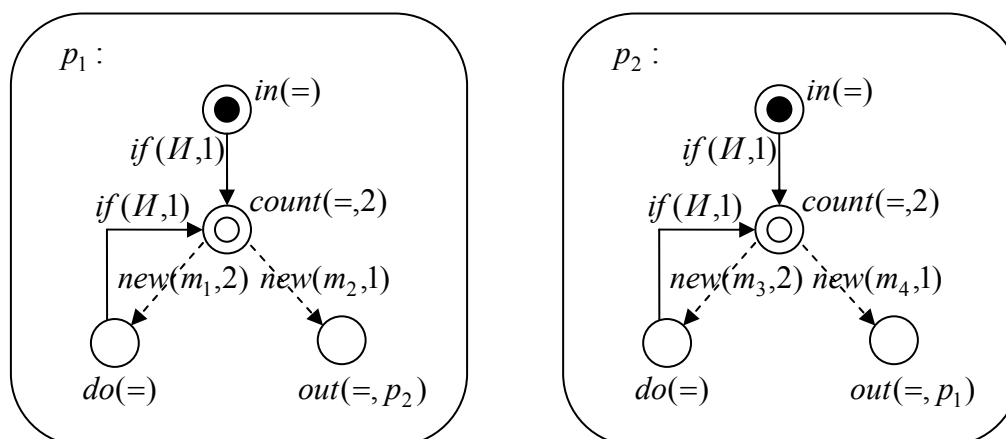


Рис. 3. Пример представления кооперативного взаимодействия процессов множеством графов Р-объектов

Пусть состояние взаимодействующих Р- и М-объектов, хранимое в переменных $p[]$ и $m[]$ модели GraphPlus, отображается на переменные алгоритма $sp, nxt, c_i, sm, ptype, mtype, node, a_{ij}$. Переменная sp является частью состояния, хранимого в переменной модели $p[]$. Она описывает особенности моделируемого вычислительного процесса. Переменная nxt хранит идентификатор Р-объекта, который будет являться значением функции $next$ модели GraphPlus. С каждой вершиной графа Р-объекта, помеченной меткой $count()$, связывается специальная переменная-счетчик c_i . Индекс переменной c_i соответствует числовой пометке вершины. Переменные c_i принимают значения в диапазоне от 0 до числа в метке $count()$. Эти значения сохраняются между последовательными посещениями. Переменная sm является частью состояния, хранимого в переменной модели $m[]$. Она, аналогично sp , описывает особенности моделируемого вычислительного процесса. Переменная $mtype$ принимает значение «посещающего» М-объекта. Переменная $ptype$ принимает значение «посещаемого» Р-объекта. Переменная $node$ хранит числовую метку текущей вершины на графе Р-объекта. Номер вершины, с которой начнется выполнение алгоритма, определяется тем, был ли М-объект только что активирован или переходит из некоторого Р-объекта по функции $next$. Переменные a_{ij} принимают значения Р-объекта, в который направляется активированный М-объект. Каждая из переменных a_{ij} ставится в соответствие дуге на графе Р-объекта, имеющей пометку $new()$ и обозначающей активируемый локальный М-объект. В начале каждого посещения все переменные a_{ij} имеют значение ε . Значения переменных a_{ij} в конце посещения являются вычисленными значениями функции $activate$ модели GraphPlus. Первый аргумент функции $activate$ соответствует М-объекту, указанному в пометке $new()$, у дуги, с которой сопоставлена переменная a_{ij} . Значение переменных a_{ij} вычисляется заново при каждом посещении. Пусть также имеется вспомогательная функция $init_node$. Данная функция для М-объекта определяет номер вершины, с которой, будучи активированным, начнет посещение графа Р-объекта рассматриваемый М-объект. Эта вершина однозначно определяется из диаграммы Р-объекта: в нее ведет дуга с меткой new , содержащей этот М-объект.

С учетом введенных обозначений классам пометок сопоставляются семь фрагментов алгоритма «посещения».

(* ПОМЕТКА : $in(X)$, ВЕРШИНА 1 *)

ШАГ 1 : ДЛЯ ВСЕХ $a_{i,j}$ ВЫПОЛНИТЬ $a_{i,j} := \varepsilon$

ЕСЛИ $node \neq 1$ ТО ПЕРЕЙТИ К ШАГУ $node$

ЕСЛИ < инициализация > ТО ДЛЯ ВСЕХ c_i ВЫПОЛНИТЬ $c_i := 0$

ВЫЧИСЛИТЬ $(sp, sm) := X(sp, sm)$

ПЕРЕЙТИ К ШАГУ 1.1

(* ПОМЕТКА : $out(X, Y)$, ВЕРШИНА N *)

ШАГ N : ВЫЧИСЛИТЬ $(sp, sm) := X(sp, sm)$

$nxt := Y$

$node := 1$

КОНЕЦ

(* ПОМЕТКА : $do(X)$, ВЕРШИНА N *)

ШАГ N : ВЫЧИСЛИТЬ $(sp, sm) := X(sp, sm)$

ПЕРЕХОД К ШАГУ $N.1$

(* ПОМЕТКА : $count(X, Y)$, ВЕРШИНА N *)

ШАГ N : ВЫЧИСЛИТЬ $(sp, sm) := X(sp, sm)$

$c_N := c_N + 1$

ЕСЛИ $c_N = Y$ ТО

$c_N := 0$

ДЛЯ ВСЕХ дуг (N, i) с меткой $new()$ ВЫПОЛНЯТЬ $a_{N,i} := ptype$

КОНЕЦ ЕСЛИ

ЕСЛИ $L(ptype, mtype) \wedge a_{N(ptype), N(mtype)} \neq \varepsilon$ ТО $nxt := ptype$

ИНАЧЕ $nxt := \varepsilon$

$node := init_node(mtype)$

КОНЕЦ

(* ПОМЕТКА : $ift(X, Y)$, ДУГА : (N, M) *)

ШАГ $N.Y$: ЕСЛИ $mtype = X$ ТО ПЕРЕХОД К ШАГУ M

ИНАЧЕ ПЕРЕХОД К ШАГУ $N.(Y + 1)$

(* ПОМЕТКА : $if(X, Y)$, ДУГА : (N, M) *)

ШАГ $N.Y$: ЕСЛИ $X(sp, sm)$ ТО ПЕРЕХОД К ШАГУ M

ИНАЧЕ ПЕРЕХОД К ШАГУ $N.(Y + 1)$

(* ПОМЕТКА : $new(X, Y)$, ДУГА : (N, M) *)

ШАГ $N.Y$: $a_{N,M} := ptype$

ПЕРЕХОД К ШАГУ $N.(Y + 1)$

Очевидным образом из перечисленных фрагментов для произвольного графа P -объекта можно построить конкретный алгоритм «посещения». Таким образом, множеству размеченных графов P -объектов соответствует алгоритм, вычисляющий функциональные отношения модели $activate$, $next$, $newP$, $newM$. Поэтому, с учетом описания начального состояния вычислений, графическое представление модели GraphPlus эквивалентно ее точному описанию (1) в терминах темпоральной логики.

В третьей главе рассмотрен синтаксис языка моделирования пространственно-распределенных дискретных систем GraphPlus. Для выполнения анализа моделей процессов методом имитационного

моделирования, а также построения кода каркасных библиотек требуется метод машинно-ориентированного представления моделей. Метод основан на графической форме спецификации Р-объектов, пример которой представлен на рис. 2.

Математическое описание разработанного языка моделирования выполнено с использованием Расширенных Бэкуса-Наура Форм (РБНФ). Сохраняя семантику диаграмм процессов (описываемую алгоритмом «посещения» в главе 2) язык моделирования GraphPlus вводит следующее добавление. Предполагается, что граф из Р-объектов (рис. 1) для произвольной модели имеет вид дерева. Тогда, добавив новую метку в язык диаграмм для ссылки на диаграмму Р-объекта на более низком уровне иерархии, и введя систему типов Р- и М-объектов, мы получим инструмент для моделирования сложного вычислительного процесса.

Дополнительно для возможности компактного описания сложных систем Р-объектов в язык моделирования GraphPlus введен механизм модулей с параметрами:

```

<module> ==
  MODULE <ident> [PARAM <ident>{,<ident>}]
    [COMMENT <str>]
    { <import>|
      <type>|<var>|
      <job>|<proc_ref>|
      <action>|<condition>|
      <port>
    }
  END.

<import> ==
  IMPORT <ident> [AS <ident>]
    [PARAM <ident>{,<ident>}].

```

Его применение иллюстрируется при описании каркасов распределенных алгоритмов в главе 4. Для связи с языком реализации используются предложения <action> (процедура на языке реализации); <condition> (логическая функция на языке реализации); <port> (взаимодействие с окружением для ввода-вывода). Предложение <var> обозначает переменные численного метода, а <type> — их тип в языке реализации. Терминами **PROCESS** и **JOB** обозначены классы Р- и М-объектов.

Граф Р-объекта описывается как последовательность предложений <node>:

```

<process> ==
  PROCESS <id_def> ENTRY <label>
    [COMMENT <str>]
    {<proc_com>|<node>}
  END.

<node> ==
  <label> (CALL <id_use> | STUB)
    [EXIT [<num>]]
    [PIC <num> " ," <num>]
  {

```

```
(IF <id_use> THEN | CONTINUE) <label>
  [PIC <num> "," <num>] |
(NEW <id_use> TO <label>
  [PIC <num> "," <num>])
```

}.

Каждое предложение обозначает вершину графа и список дуг, исходящих из вершины. Координатная привязка вершин и дуг к плоскости диаграммы выполняется путем указания координат после слова `pic`. Координата дуги обозначает положение ее изгиба. Текст описания Р-объекта можно формировать и отображать с помощью специального графического редактора, входящего в программный комплекс моделирования.

В главе 3 подробно описывается язык моделирования GraphPlus с примерами использования синтаксических конструкций. Также рассмотрены особенности синтаксического анализа на основе дополнительной формы представления моделей — машинно-ориентированного описания в виде сети структурных элементов. Этот способ представления эквивалентен текстовой и графической форме и предназначен для трансляции и анализа.

В четвертой главе на основе языка моделирования GraphPlus построены библиотеки численных методов, использующие графическое объектное представление схем параллельных алгоритмов.

На примере простейшей схемы **MAP** «применить ко всем» рассмотрен принцип описания каркасной библиотеки, то есть семейства алгоритмов с общим способом организации вычислений.

Описание состоит из следующих частей. Вначале приводится неформальное объяснение принципа работы схемы. Так схема **MAP** образована из n независимых Р-объектов F , каждый из которых вычисляет функцию $f()$ для своего значения аргумента x_i . Вычисления заканчиваются, когда все объекты F вычислят свои $f(x_i)$. Затем записывается алгоритм для эквивалентной последовательной схемы, который в случае **MAP** имеет вид (10). Дополнительно могут быть указаны ограничения, выраженные через абстрактные функции и предикаты последовательной схемы, используемые при ее распараллеливании.

```
ДАНО  $x_i : i = \overline{1..n}$ 
НАЙТИ  $y_i : i = \overline{1..n}$ 
ДЛЯ ВСЕХ  $i = \overline{1..n}$  ВЫПОЛНЯТЬ
   $y_i := f(x_i)$ 
КОНЕЦ
```

(10)

Далее следует определение структуры схемы. Оно представляет собой набор параметров, часть из которых задает масштаб (максимальное число параллельных процессов в реализации), а другая часть является определением специальных подклассов Р- и М-объектов модели вычислений. Подклассы Р- и М-объектов конкретизируются в терминах функций, предикатов и переменных, использованных в последовательной схеме. Для схемы **MAP** определение структуры описывается формулой (11).

$$\begin{aligned}
 A &= A_{MAP}(n, F), A_{MAP} \subset T, n \in \mathbb{N} \\
 F &= F(x, y | f)\{y' = f(x)\}, F \subset P
 \end{aligned}
 \tag{11}$$

Формула (11) описывает множество реализаций параллельных алгоритмов A , использующих предлагаемую вычислительную модель на основе Р- и М-объектов и распараллеливающих последовательный алгоритм (10). Алгоритмы по схеме **MAP** принадлежат к множеству всех реализуемых моделью алгоритмов T . Схема имеет единственный параметр масштаба n , который задает количество Р-объектов. Параметр F определяет класс Р-объектов схемы. Формула (11) показывает, что каждый Р-объект схемы F содержит переменные объекта x и y . За вертикальной чертой в списке параметров объекта F следует список функций, связанных с переменными объекта. В данном случае это функция f . В определении функций объекта, которое заключено в фигурные скобки, могут присутствовать произвольные переменные, а также переменные состояния объекта, перечисленные в списке параметров. Если связанная с состоянием объекта переменная употребляется в списке вычисляемых параметров функции, то это новое значение переменной объекта обозначается при помощи штриха (').

Определение (11) позволяет воспользоваться готовой библиотекой для реализации параллельных вычислений, эквивалентных алгоритму (10). Для этого требуется определить f , x , y , расширяя библиотечный класс F (определив пользовательский подкласс F' класса F). Однако определение (11) скрывает способ реализации параллельного алгоритма. Реализация схемы **MAP** в терминах модели Р- и М- объектов может быть формализована на языке GraphPlus следующим образом.

```

MODULE MAIN
  PROCESS MAIN
  IMPORT STUFF AS stuff
  IMPORT MAP AS map
  IMPORT rMAP AS M4      PARAM map, stuff
  IMPORT rMAP AS M16     PARAM M4,  stuff
  IMPORT rMAP AS M64     PARAM M16, stuff
  IMPORT rMAP AS M256    PARAM M64, stuff
  IMPORT rMAP AS M1024   PARAM M256, stuff
END

PROCESS MAIN ENTRY @begin
  @begin CALL F FROM M1024
END

-----
MODULE STUFF
  JOB o EXPORT
END
-----
MODULE rMAP PARAM Tr, S
  PROCESS F; -- диаграмма
END
-----
MODULE MAP
  PROCESS F

```

```

ACTION f PARAM x IN, y OUT END
VAR x TYPE X
VAR y TYPE Y
TYPE X TYPE Y
END

PROCESS F EXPORT ENTRY @begin
  @begin CALL f
END

```

Описание алгоритма содержится в четырех модулях: **MAIN**, **STUFF**, **MAP** и **rMAP**. Назначение модуля **MAIN** заключается в формировании дерева Р-объектов нужного размера методом рекурсивного импорта модулей. Шаг рекурсии определяет модуль **rMAP**. В случае, когда первый параметр этого модуля связан с модулем типа **MAP**, например, в модуле **M4**, модуль **rMAP** представляет четыре Р-объекта. В случае, когда первый параметр модуля **rMAP** связан с модулем типа **rMAP**, представляющим n Р-объектов, модуль **rMAP** представляет $4n$ Р-объектов. Таким образом, число Р-объектов в нашей модели экспоненциально зависит от числа предложений импорта и может быть достаточно большим. Число Р-объектов в описываемом примере равно 1024. Процесс **F** из модуля **rMAP** описывается диаграммой, показанной на рис. 4.

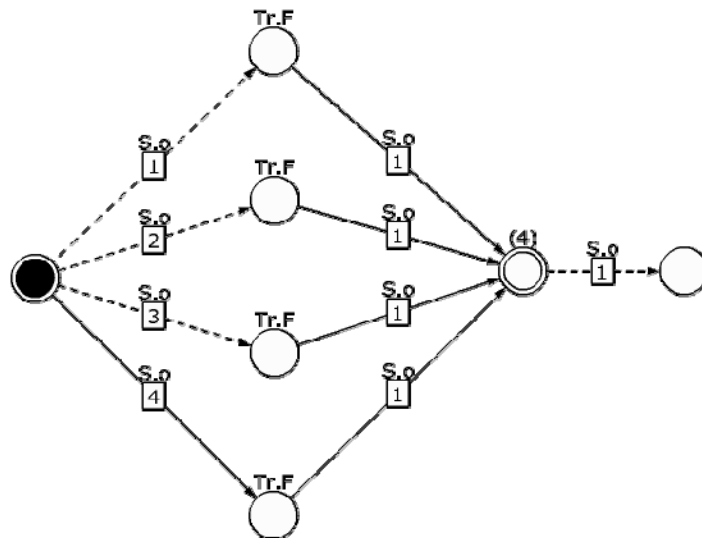


Рис. 4. Диаграмма процесса **F** из модуля **rMAP** схемы «применить ко всем»

Разработанная библиотека каркасов также включает схемы **TASKBAG** и **CHAIN**. Схема **TASKBAG** «портфель задач» описывает вычислительный процесс, в котором участвует Р-объект B , отвечающий за выдачу заданий, и группа Р-объектов T , обрабатывающих задания. При этом запрос новых заданий и возвращение обработанных (в М-объектах M_2, M_1) организуется асинхронно по готовности. Последовательная схема алгоритма вычислений имеет вид (12). Для ее корректного распараллеливания должно выполняться ограничение $p(p(r, y_1), y_2) = p(p(r, y_2), y_1)$.

ДАНО s – исследуемое множество

НАЙТИ r – множество результатов

ПОКА $\neg e(s)$ ВЫПОЛНЯТЬ

$$(s, x) := g(s) \quad (12)$$

$$y := f(x)$$

$$r := p(r, y)$$

КОНЕЦ

Параллельная декомпозиция схемы (12) представлена выражением (13).
 Диаграмма процесса верхнего уровня схемы «портфель задач» на языке моделирования GraphPlus показана на рис. 5.

$$\begin{aligned} A &= A_{TASKBAG}(n, B, T, M_1, M_2), A_{TASKBAG} \subset T, n \in \mathbb{N} \\ B &= B(s, r | g, p, e) \{ (s', x) = g(s), r' = p(r, y), e(s) \in \{0, 1\} \}, B \subset P \\ T &= T(f) \{ y = f(x) \}, T \subset P \\ M_1 &= M_1(a_x | m_x, \overline{m_x}) \{ a'_x = m_x(x), x = \overline{m_x}(a_x) \}, M_1 \subset M \\ M_2 &= M_2(a_y | m_y, \overline{m_y}) \{ a'_y = m_y(y), y = \overline{m_y}(a_y) \}, M_2 \subset M \end{aligned} \quad (13)$$

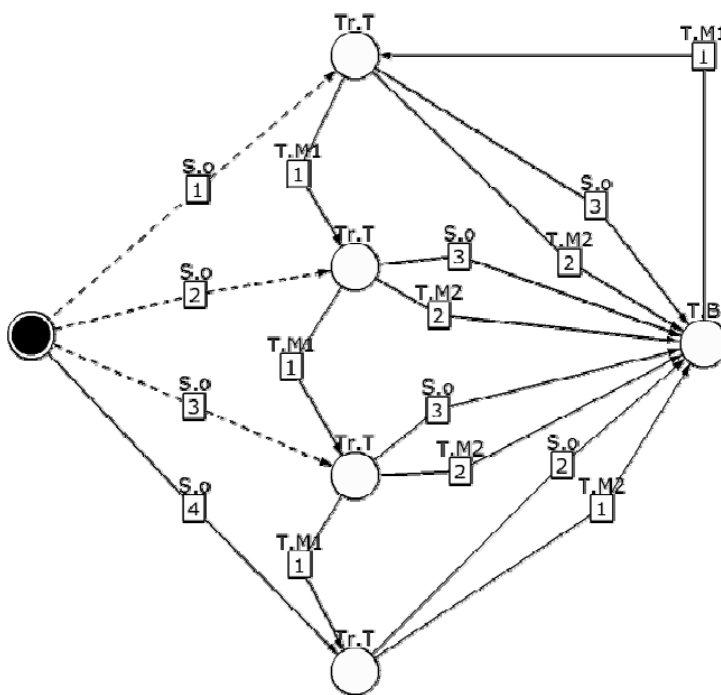


Рис. 5. Диаграмма процесса верхнего уровня схемы «портфель задач»

$$\begin{cases} \dot{N} = (BM - A)N \\ \dot{n} = (bM - a)n + D\Delta n \\ \dot{M} = Q - GM - CN - cn + f(\bar{r}, t) \end{cases} \quad (14)$$

Применение схемы «портфель задач» проиллюстрировано на примере задачи аппроксимации интеграла непрерывной функции методом адаптивной квадратуры. В качестве задачи, имеющей прикладное значение, рассмотрено сканирование параметрического пространства для модели Лотки-Вольтерра с диффузионным и стохастическим членами (14), применяемой для исследования процессов межвидовой конкуренции. Для одномерного пространства требуется найти частные решения системы (15) в стационарном состоянии при разных значениях параметров. Особенности поведения модели являются большая вычислительная сложность и зависимость времени достижения стационарного состояния от текущих параметров. Схема «портфель задач» при данных особенностях обеспечивает балансировку загруженности и скрывает интерфейс использованной для управления вычислениями пакетной системы Condor.

$$\begin{aligned}
N_i^j &= \tau(M_i^{j-1} - 1)N_i^{j-1} + N_i^{j-1} \\
n_{i-1}^j - n_i^j \left(2 + \frac{Ah^2}{D\tau\sigma}\right) + n_{i+1}^j &= \\
&= -\frac{(1-\sigma)}{\sigma} \left(n_{i+1}^{j-1} - 2n_i^{j-1} + n_{i-1}^{j-1}\right) - \frac{Ah^2}{D\tau\sigma} n_i^{j-1} - \frac{Ah^2}{D\tau\sigma} n_i^{j-1} \left(\frac{b}{B} M_i^{j-1} - \frac{a}{A}\right) \\
M_i^j &= \tau \left(\frac{QB}{A^2} - \frac{G}{A} M_i^{j-1} - \frac{B}{A^2} \left(Q - G \frac{A}{B}\right) N_i^{j-1} - \frac{c}{C} \frac{B}{A^2} \left(Q - G \frac{A}{B}\right) n_i^{j-1} + \frac{B}{A^2} f_i^{j-1}\right) + M_i^{j-1}
\end{aligned} \tag{15}$$

Схема «цепь асинхронно взаимодействующих процессов» **CHAIN** описывает вычисления на цепи попарно связанных Р-объектов L , C и R (L замыкает цепь слева, R — справа). При вычислениях происходит перенос информации в оба направления по цепи М-объектами R_m и L_m . Последовательная схема алгоритма вычислений имеет вид (16).

ДАНО $l_i, v_i, r_i : i = \overline{1..n}$; t
 НАЙТИ l_i, v_i, r_i
 ВЫПОЛНЯТЬ t РАЗ
 ДЛЯ ВСЕХ $i = \overline{1..n}$ ВЫПОЛНЯТЬ
 ЕСЛИ $i = 1$ ТО $l_1 := b_1(l_1, v_1)$ ИНАЧЕ $l_i := s_1(r_{i-1}, l_i, v_i)$
 $v_i := c_1(l_i, v_i, r_i)$
 ЕСЛИ $i = n$ ТО $r_n := b_2(v_n, r_n)$ ИНАЧЕ $r_i := s_2(v_i, r_i, l_{i+1})$
 КОНЕЦ i
 КОНЕЦ

Параллельная декомпозиция схемы (16) представлена выражением (17). Пример диаграммы процесса для схемы «цепь асинхронно взаимодействующих процессов» на языке моделирования GraphPlus приведен на рис. 6.

$$\begin{aligned}
 A &= A_{CHAIN}(n, t, L, C, R, R_m, L_m), A_{CHAIN} \subset T, n \in \mathbb{N}, t \in \mathbb{N} \\
 L &= L(l_1, v_1, r_1, l_2 | b_1, c_1, s_2) \{l'_1 = b_1(l_1, v_1), v'_1 = c_1(l_1, v_1, r_1), r'_1 = s_2(v_1, r_1, l_2)\}, L \subset P \\
 C &= C(r_1, l_2, v_2, r_2, l_3 | s_1, c_1, s_2) \{l'_2 = s_1(r_1, l_2, v_2), v'_2 = c_1(l_2, v_2, r_2), r'_2 = s_2(v_2, r_2, l_3)\}, C \subset P \\
 R &= R(r_2, l_3, v_3, r_3 | s_1, c_1, b_2) \{l'_3 = s_1(r_2, l_3, v_3), v'_3 = c_1(l_3, v_3, r_3), r'_3 = b_2(v_3, r_3)\}, R \subset P \\
 R_m &= R_m(a_r | m_r, \overline{m_r}) \{a'_r = m_r(r), r = \overline{m_r}(a_r)\}, R_m \subset M \\
 L_m &= L_m(a_l | m_l, \overline{m_l}) \{a'_l = m_l(l), l = \overline{m_l}(a_l)\}, L_m \subset M
 \end{aligned} \tag{17}$$

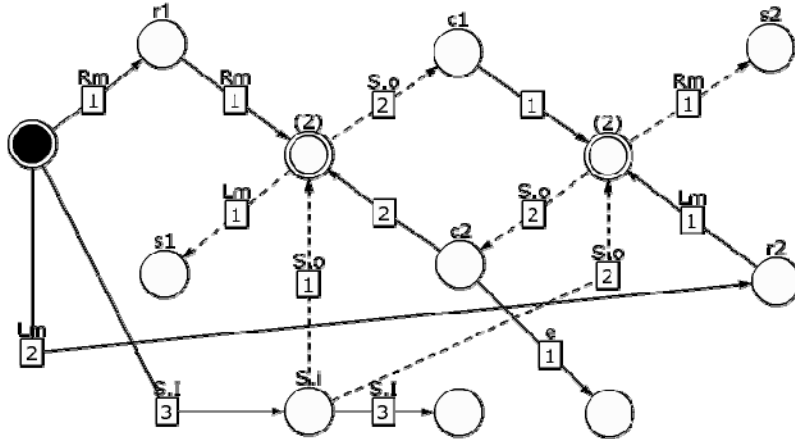


Рис. 6. Диаграмма процесса \mathbf{C} схемы «цепь асинхронно взаимодействующих процессов»

Применение схемы «цепь асинхронно взаимодействующих процессов» проиллюстрировано на задаче о распараллеливании решения уравнения Лапласа на основе метода Гаусса-Зейделя. В качестве задачи, имеющей прикладное значение, описан способ параллельного моделирования распространения световых волн в диэлектрике. В работе Головашкина Д.Л. и Сойфера В.А.²⁾ предложена математическая модель распространения световой волны в диэлектрике, применение которой заключается в решении системы уравнений вида (18). Особенностью решения является необходимость итеративного пересчета трех матриц проекций электрического и магнитного полей, что требует значительных вычислительных мощностей.

$$\left\{ \begin{aligned}
 H_z^{n+1} &= \frac{h_t}{\mu_0 \mu} \frac{E_x^{n+1} - E_{x_{j-1}}^{n+1}}{h_y} + H_z \\
 E_x^{n+1} &= \frac{h_t}{\varepsilon_0 \varepsilon} \left(\frac{H_{z_{j+1}} - H_z}{h_y} - \frac{H_{y_{k+1}}^{n+1} - H_y^{n+1}}{h_z} \right) + E_x \\
 H_y^{n+1} &= -\frac{h_t}{\mu_0 \mu} \frac{E_x^{n+1} - E_{x_{k-1}}^{n+1}}{h_z} + H_y
 \end{aligned} \right. \tag{18}$$

В диссертации показано, что алгоритм решения системы (18), аналогично типовому методу Гаусса-Зейделя, удастся свести к последовательной схеме (16), а затем к параллельной схеме (17).

²⁾ Головашкин Д.Л., Сойфер В.А. Анализ прохождения электромагнитного излучения через дифракционную линзу // Автометрия. 1999. Вып.6. С.119-121.

В пятой главе рассмотрена реализация программного комплекса и алгоритмов численного моделирования, экспериментально обоснована эффективность предложенных методов управления вычислениями.

Инструментальная среда моделирования GraphPlus включает графический редактор, компилятор, кодогенератор, интерпретатор, варианты среды времени исполнения для имитационных экспериментов и многопоточного исполнения, службу управления вычислениями в системе Condor.

Предложен метод исполнения моделей, представленный в виде паттерна проектирования, названного «Постоялец-Посетитель». Метод заключается в выделении специальных объектов **Room**, связанных с потоками управления, перестраивающих списковые структуры из объектов **Resident** и **Visitor**, являющихся родовыми для P- и M- объектов модели GraphPlus. Метод проверен нагрузочными испытаниями на разных типах компьютеров. Таблица 1 иллюстрирует высокую эффективность данного решения.

Таблица 1

Число сегментов		Тип рабочей станции					
		Однопроцессорная ¹⁾		Однопроцессорная гиперпоточная ²⁾		Двухпроцессорная ³⁾	
		Время, с	Ускор.	Время, с	Ускор.	Время, с	Ускор.
Оптимизированный код	2	2,1050	1,0024	1,4171	1,5393	1,7406	1,9668
	50	2,1661	0,9710	1,4516	1,5026	1,8172	1,8830
Каркас	2	2,1120	0,9996	1,4203	1,5358	1,7516	1,9571
	50	2,1431	0,9846	1,4781	1,4758	1,8672	1,8359

Примечание: ускорение вычислялось относительно длительности счета эквивалентной последовательной программы, результаты усреднены по 10 реализациям; использовались следующие конфигурации:

- 1) CPU AthlonXP 1700+ 1.47 GHz RAM 256MB OS MS Windows XP 2002 SP2;
- 2) CPU Pentium4 3.4 GHz RAM 2GB OS MS Windows XP 2002 SP2;
- 3) CPU PentiumIIIx2 1 GHz RAM 512MB OS MS Windows 2000 SP4.

В диссертации также приведены результаты нагрузочного тестирования системы Condor, работающей под управлением сервера объектов модели GraphPlus для организации распределенного исполнения по схеме **TASKBAG**. Тесты показали, что метод управления вычислениями целесообразен, когда общее количество заданий велико, и нет возможности использования кластера в выделенном режиме весь период расчета. В данном случае преимуществом, компенсирующим невысокую эффективность (порядка 70% для 1 минуты автономного счета), является автоматическая поддержка непрерывных вычислений при сбоях, выключении питания, использовании компьютеров локальными привилегированными пользователями.

Далее в главе 5 описан метод проведения имитационных экспериментов с использованием модели GraphPlus для вычислительной среды с флуктуирующими характеристиками производительности и пропускной способности. Предложен метод измерения и учета влияния фоновых процессов операционной системы и коммуникационной среды на разбалансировку вычислений. Выполнено подробное исследование вычислительного процесса

схемы **CHAIN** имитационным методом. В результате были построены пространственно-временные диаграммы, показывающие, что схема **CHAIN** позволяет совместить вычисления и обмен данными. Изучены виды переходных процессов (рис. 7) при различных интенсивностях случайного воздействия на систему процессов.

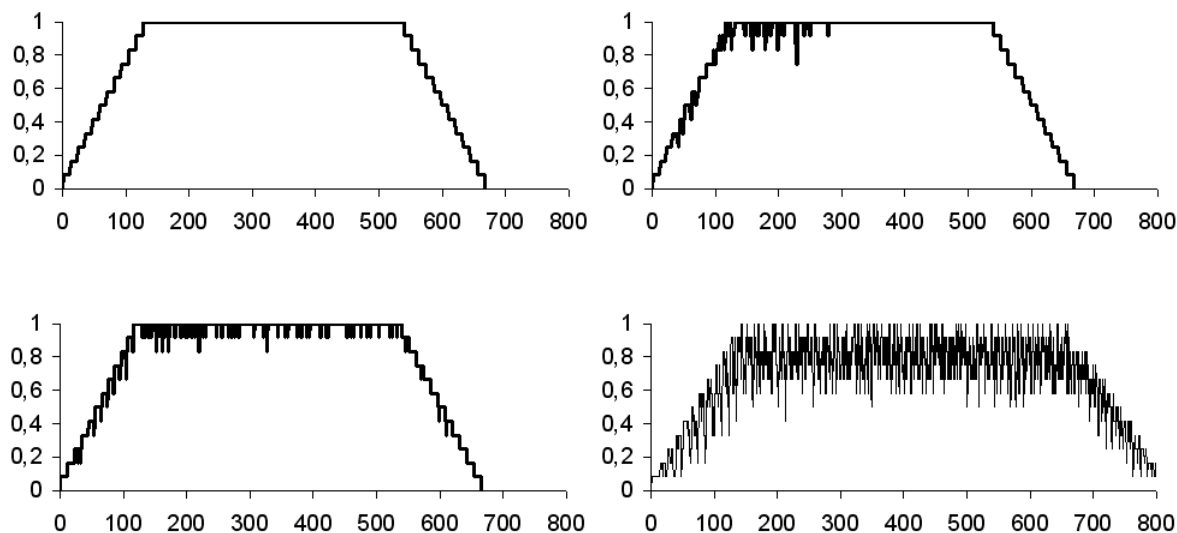


Рис. 7. Изменение эффективности вычислений в модельном времени при различных типах возмущающего воздействия

Получен важный результат: у вычислительной модели **CHAIN** имеется «интервал устойчивости», в котором как величина, так и характер распределения времен задержек не оказывают существенного влияния на эффективность. Этому эффекту соответствует тип поведения, показанный на первых трех графиках рис. 7. Таким образом, модель **CHAIN** применима для распределенных вычислений в гетерогенных средах с флуктуирующими параметрами эффективности и пропускной способности.

Также проведено исследование влияния флуктуации производительности, вызванной фоновыми процессами операционной системы. Показано, что фоновые процессы приводят к снижению эффективности примерно на 2% внутри «интервала устойчивости». После увеличения времени передачи сообщений более критического, когда происходит частичный простой Р-объектов, влияние флуктуации производительности стремится к нулю.

Выполненные имитационные эксперименты позволяют сделать заключение о применимости описанного в работе метода построения библиотек численных методов по моделям вычислительных процессов. Кроме этого, эксперименты иллюстрируют методику проверки эффективности для вновь проектируемой каркасной библиотеки путем дискретно-событийной имитации, не прибегая к дорогим вычислительным экспериментам или трудоемкому аналитическому исследованию. При этом исследуемый прикладной код не требует существенных переделок, а заменяется лишь сервер объектов модели GraphPlus.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

1. Предложена новая графическая объектная модель параллельных и распределенных вычислений GraphPlus. С использованием темпоральной логики Лампорта построена спецификация модели GraphPlus для представления вычислительных процессов произвольного вида в форме удобной для реализации в распределенных гетерогенных вычислительных средах.

2. На основе интерпретации формул для моделей конкурентного и кооперативного взаимодействия процессов, являющихся типовыми при реализации вычислительных алгоритмов, показана адекватность модели GraphPlus при описании параллельных вычислительных алгоритмов для распределенных и гетерогенных вычислительных сред.

3. Определен способ наглядного изображения моделей процессов в виде множества ориентированных помеченных графов с семантикой, соответствующей представлению в виде темпоральной формулы. По графическому описанию модели GraphPlus построен метод порождения алгоритмов для вычисления функций, определяющих динамические свойства модели.

4. Формализован синтаксис языка моделирования пространственно-распределенных дискретных систем, включающий текстовую и графическую нотации для описания структуры и логики работы процессов. Предложен метод машинно-ориентированного описания моделей пространственно-распределенных дискретных систем, используемый в алгоритмах программного комплекса.

5. Разработано представление схем параллельных вычислительных процессов для реализации прикладных алгоритмов численного моделирования, включающее определение семантики в форме схемы последовательного алгоритма; способ отделения численного метода от реализующего алгоритма; параллельную схему алгоритма вычислений в терминах объектной модели на языке GraphPlus.

6. Реализованы схемы процессов MAP «применить ко всем», TASKBAG «портфель задач» и CHAIN «цепь асинхронно взаимодействующих процессов». С использованием разработанных схем рассмотрены примеры распараллеливания тестовых, а также практически значимых задач математического моделирования, имеющих высокую вычислительную сложность, таких как задача моделирования процесса межвидовой конкуренции и задача моделирования распространения света на основе решения уравнений Максвелла.

7. Разработан программный комплекс моделирования. Проведена экспериментальная проверка эффективности компонента программного комплекса, отвечающего за исполнение произвольного вычислительного процесса, соответствующего спецификации модели GraphPlus. В сравнительных тестах продемонстрирована высокая эффективность исполнения моделей на разных типах современных многопроцессорных систем с общей памятью. Продемонстрирована эффективная работа механизма исполнения

модели в распределенной вычислительной среде на примере виртуального кластера под управлением системы Condor.

8. Предложен метод исследования работы вычислительного приложения на аппаратуре с нестабильными характеристиками производительности и пропускной способности коммуникационной среды. Метод основан на использовании специальной среды времени исполнения моделей, представляющей собой монитор дискретно-событийного моделирования с механизмом продвижения времени, основанном на событиях. Также предложены методики измерения и имитации параметров вычислительной среды при проведении имитационных экспериментов. Проведены детальные исследования каркасной библиотеки CHAIN методом имитации гетерогенной асинхронной вычислительной среды с различными параметрами, демонстрирующими пригодность библиотеки для данного типа вычислений.

В целом полученные результаты можно рассматривать как методическую основу для построения каркасных библиотек типовых вычислительных процессов, их имитационного моделирования и реализации в произвольных вычислительных средах, на разных языках программирования, с использованием различных прикладных интерфейсов для управления параллельным протеканием и взаимодействием процессов. Таким образом, в работе решена крупная научная проблема, имеющая важное практическое значение.

Основное содержание работы полностью раскрывается в следующих публикациях:

Монографии:

1. Востокин С.В. Графическая объектная модель параллельных процессов и ее применение в задачах численного моделирования. Самара: Издательство Самарского научного центра РАН, 2007. 286 с. ISBN 978-5-93424-284-9.
2. Востокин С.В. Метод описания пространственно-распределенных параллельных процессов. Самара: Издательство Самарского научного центра РАН, 2004. 86 с. ISBN 5-93424-141-9.
3. Востокин С.В. Спецификация языка программирования ГРАФ ПЛЮС. Самара: Издательство Самарского научного центра РАН, 2003. 86 с. ISBN 5-93424-101-X.

Публикации в периодических изданиях из списка ВАК:

4. Востокин С.В. Спецификация модели параллельных и распределенных вычислений GraphPlus на основе логики TLA // Известия Самарского научного центра РАН. 2006. Т.8, №3(17). С.875-881.
5. Востокин С.В. Особенности реализации процедур сканирования параметрических пространств для сложных численных моделей // Математическое моделирование. 2006. Т.18, №12. С. 125-128.
6. Востокин С.В. Технология визуального проектирования параллельных и распределенных приложений // Системы управления и информационные технологии. 2006. №2 (24). С.39-43.

7. Востокин С.В. Объектно-ориентированный метод структурирования кода метакомпьютерного приложения // Информационные технологии. 2006. №5. С.40-45.
 8. Востокин С.В. Инструментальная среда метакомпьютерного проекта GraphPlus // Вестник Самарского государственного аэрокосмического университета им. С.П. Королева. 2006. №3(11). С.60-65.
 9. Востокин С.В. Применение метода парного взаимодействия объектов для построения сред разработки распределенных приложений // Вестник Самарского государственного технического университета. Серия: Технические науки. 2005. №38. С.26-28.
 10. Востокин С.В., Курушина С.Е. Численное моделирование процесса конкуренции во флуктуирующих средах на кластерных вычислительных системах // Известия Самарского научного центра РАН. 2005. Т. 7, №1(13). С.143-148.
 11. Востокин С.В. Применение интерпретатора сценария GraphPlus для управления распределенными вычислениями // Известия Самарского научного центра РАН. 2005. Т. 7, №1(13). С.138-142.
 12. Востокин С.В., Прохоров С.А. Пример визуальной модели распределенного вычислительного процесса // Известия Самарского научного центра РАН. 2004. Т. 6, №1(10). С.180-185.
 13. Востокин С.В. Технология моделирования распределенных систем, основанная на визуальном языке, и ее приложения // Известия Самарского научного центра РАН. 2004. Т. 6, №1(10). С.185-193.
 14. Востокин С.В. Графический метод проектирования параллельных программ с использованием асинхронной событийной модели вычислений // Вестник Самарского государственного технического университета. 2004. №30. С.178-183.
 15. Востокин С.В. Модель параллельных вычислений визуального графо-ориентированного языка // Сб. Вестник Самарского государственного аэрокосмического университета им. С.П. Королева, Серия: Актуальные проблемы радиоэлектроники. 2000. Вып. 4. С.96-100.
 16. Востокин С.В. Автоматизация параллельного программирования с использованием визуального языка // Сб. Вестник Самарского государственного аэрокосмического университета им. С.П. Королева, Серия: Актуальные проблемы радиоэлектроники. 2000. Вып. 1. С.80-83.
- Свидетельства о регистрации программ для ЭВМ:*
17. Востокин С.В. Модуль многопоточного исполнения моделей программного комплекса проекта Граф Плюс. Свидетельство об официальной регистрации программы для ЭВМ №2007611206.
 18. Востокин С.В. Транслятор моделей программного комплекса проекта Граф Плюс. Свидетельство об официальной регистрации программы для ЭВМ №2007611201.
- Другие публикации:*
19. Востокин С.В. Технология визуального программирования каркасов параллельных и распределенных приложений на основе объектно-ориентированной модели вычислений GraphPlus // Труды III

- Международной конференции «Параллельные вычисления и задачи управления». М.: ИПУ РАН им. Трапезникова, 2006. С. 634-646.
20. Востокин С.В. Каркас «пакет с заданиями» для программирования grid-приложений поисково-переборного типа на базе системы Condor // Распределённые вычисления и грид-технологии в науке и образовании: тез. докл. 2-й междунар. конф. Дубна: ОИЯИ, 2006. С.146-147.
 21. Востокин С.В. Объектная модель для разработки параллельных вычислительных приложений // Математическое моделирование и краевые задачи: труды третьей всероссийской научной конференции. Самара: изд-во СамГТУ, 2006. Ч.4 С.36-39.
 22. Востокин С.В. Исследование связи между эффективностью распределённых вычислений и качеством обслуживания коммуникационной среды с использованием имитационной модели // Надежность и качество. Пенза: Инф.-изд. центр Пенз. гос. ун-та, 2006. С.225-226.
 23. Востокин С.В. Визуальное проектирование каркасов параллельных и распределённых приложений в технологии GraphPlus // Перспективные информационные технологии в научных исследованиях, проектировании и обучении: труды научно-технической конференции с международным участием. Самара, 2006. Т.1. С. 51-58.
 24. Востокин С.В. Курушина С.Е. Применение вычислительного кластера для моделирования процесса конкуренции во флуктуирующих средах // Компьютерные технологии в науке, практике и образовании: труды всероссийской межвузовской научно-практической конференции. Самара, 2005. С.6-9.
 25. Востокин С.В. Инструментальная система проектирования GRID приложений для архитектур Master-Worker // Надежность и качество. Пенза: Инф.-изд. центр Пенз. гос. ун-та, 2005. С.243.
 26. Востокин С.В. Моделирование распределённого вычислительного процесса на визуальном языке // Информационные технологии моделирования и управления. Междунар. сборник научн. труд. Воронеж: Изд-во Научная книга, 2004. Вып.17. С.37-45.
 27. Востокин С.В. Проект ГРАФ ПЛЮС. Визуальный язык распределённого и параллельного программирования. <http://graphplus.ssau.ru>. 2004-2007.
 28. Востокин С.В. Язык моделирования пространственно распределённых параллельных процессов // Четвертый международный научно-практический семинар «Высокопроизводительные параллельные вычисления на кластерных системах». Самара, 2004. С.40-41.
 29. Востокин С.В. Технология построения распределённых приложений проекта ГРАФ ПЛЮС // Надежность и качество. Пенза: Инф.-изд. центр Пенз. Гос. ун-та, 2004. С.166.
 30. Востокин С.В. Графический метод проектирования и исследования свойств параллельных программ с использованием асинхронной событийной модели вычислений // Электронный журнал "Исследовано в России". 2004. Вып. 54. С. 613-627.

31. Востокин С.В. Визуальный язык ГРАФ ПЛЮС: свойства и цели проектирования // Математическое моделирование информационных и технологических систем: Сб. науч.тр. Воронеж: Воронеж. гос. технол. акад., 2003. Вып 6. С.150-152.
32. Vostokin S.V. Computer Aided Engineering of Distributed and Parallel Systems Using GRAPH Plus language // Interactive Systems: The Problems of Human-Computer Interaction. Proc. of the Int. Conf., 23-27 September 2003. Ulyanovsk: UISTU, 2003. P.56-59.
33. Востокин С.В. Использование визуального языка Граф Плюс для повышения надежности параллельных программ // Надежность и качество. Пенза: Инф.-изд. центр Пенз. гос. ун-та, 2003. С.182-183.
34. Востокин С.В. Метод автоматического извлечения информации из свободного текста и оценка его надежности // Надежность и качество. Пенза: Инф.-изд. центр Пенз. гос. ун-та, 2002. С.124-126.
35. Востокин С.В. Графическая модель программирования параллельных вычислений // Перспективные информационные технологии в научных исследованиях, проектировании и обучении. Самара: изд-во СГАУ, 2001. С.7-12.
36. Востокин С.В. Визуальный язык параллельного программирования системы ГРАФ // Надежность и качество. Пенза: Инф.-изд. центр Пенз. гос. ун-та, 1999. С.553-555.
37. Востокин С.В. Концепция разработки визуального языка параллельного программирования системы ГРАФ // БИКАМП-99: тез. докл. С.-Петербург, 1999. С.121.
38. Востокин С.В. Разработка многопоточных программ в системе визуального программирования ГРАФ // Математика, компьютер, образование: тез. докл. Москва, 1999. Т.6. С.59.
39. Востокин С.В. Генерация текстов программ на основе графической модели системы ГРАФ // Техн. кибернетика, радиоэлектроника и системы управления: тез. докл. Таганрог, 1998. С.105.
40. Востокин С.В. Технология генерации текстов параллельных программ по графической модели алгоритма // БИКАМП-98: тез. докл. С.-Петербург, 1998. С.57-58.
41. Востокин С.В. Компилятор подсистемы отладки системы графического программирования ГРАФ // Королевские чтения: тез. докл. Самара, 1997. С.56-57.
42. Востокин С.В. Подсистема отладки графо-символических объектов // XXIII Гагаринские чтения: тез. докл. Москва, 1997. Ч.5. С.88.