

На правах рукописи

Березовский Павел Сергеевич

**УПРАВЛЕНИЕ ЗАДАНИЯМИ В ГРИДЕ
С НЕКЛАСТЕРИЗОВАННЫМИ РЕСУРСАМИ**

Специальность 05.13.11 — Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание учёной степени
кандидата физико-математических наук

Москва — 2011

Работа выполнена в Институте прикладной математики
им. М.В. Келдыша Российской академии наук.

Научный руководитель: кандидат физико-математических наук
Коваленко Виктор Николаевич.

Официальные оппоненты: доктор физико-математических наук
Лацис Алексей Оттович,

кандидат физико-математических наук
Крюков Александр Павлович.

Ведущая организация: Объединённый институт ядерных исследований
(ОИЯИ, г. Дубна).

Защита состоится 14 июня 2011 г. в 11 часов на заседании диссертационного
совета Д 002.024.01 в Институте прикладной математики им. М.В. Келдыша
РАН по адресу: 125047, Москва, Миусская пл., 4.

С диссертацией можно ознакомиться в библиотеке Института прикладной
математики им. М.В. Келдыша РАН.

Автореферат разослан « ____ » _____ 2011 г.

Учёный секретарь диссертационного совета,
доктор физико-математических наук

Т.А. Полилова

Общая характеристика работы

Актуальность темы

За последние несколько лет широкое распространение получила концепция построения распределённой среды (инфраструктуры) *грид*. Наиболее развитым на сегодняшний день является грид вычислительного типа. В рамках этого типа основные результаты достигнуты при построении гридов из находящихся в разных точках сети кластерных узлов, в которые объединяется множество компьютеров. Такая организация вычислительной инфраструктуры получила название «*двухуровневый грид*». Тем не менее активно развивается альтернативная форма грида, где вычислительная инфраструктура строится на основе некластеризованных ресурсов (отдельных компьютеров), в том числе используемых совместно с их владельцами (*неотчуждаемых*). Формируемая подобным образом распределённая среда получила название «*одноуровневый грид*».

Одноуровневая архитектура имеет два преимущества перед двухуровневой. Первое состоит в том, что, позволяя использовать для задач грида незадействованные холостые циклы процессоров, такая архитектура предоставляет возможность создавать гриды на основе существующей ресурсной базы, в том числе из персональных компьютеров.

Второе преимущество заключается в простоте создания грид-инфраструктур. Для поддержки функционирования одноуровневого грида необходим управляющий центр — диспетчер, который существует в единственном экземпляре на всё множество пространственно распределённых компьютеров, интегрируемых в грид. Диспетчер нужен и для двухуровневого грида, но, помимо этого, в каждом ресурсном узле должна быть дополнительно установлена система управления кластером и службы доступа из грида. В одноуровневом гриде от владельцев компьютеров-ресурсов требуется лишь установка компактного и просто конфигурируемого программного обеспечения, после чего компьютер может использоваться для обработки задач грида. В обоих случаях при создании и поддержке функционирования грида необходим профессиональный подход, но в одноуровневой архитектуре не требуется наличия специального персонала, который бы занимался обслуживанием исполнительных ресурсов в связи с их включением в грид.

Рассматривая одноуровневый грид как широко доступное средство дистанционного использования вычислительных ресурсов, можно указать несколько сценариев его применения.

1. Создание распределённых инфраструктур высокой пропускной способности. Одноуровневый грид сохраняет возможность создания широкомасштабных грид-инфраструктур, которые в современной практике строятся на основе локальной кластеризации ресурсов. Необходимо, однако, отметить ограничения на класс задач, которые связаны с неотчуждаемостью ресурсов: становится невозможной обработка параллельных

(многопроцессорных) приложений ввиду того, что процессы, находящиеся на разных компьютерах, будут выполняться дискретно — в периоды малой активности их владельцев — и несогласованно друг с другом. По той же причине полное время обработки обычного задания на исполнительном компьютере не совпадает с требуемым чистым процессорным временем. В связи с этим одна из наиболее важных проблем, решаемых программными средствами, — минимизация нарушения выполнения задания в заданное время.

В таких условиях наиболее привлекательным выглядит применение масштабных одноуровневых гридов для выполнения серийных расчётов в виде набора независимых заданий, которые могут обрабатываться параллельно на разных ресурсах, не обмениваясь данными. В этом смысле их можно рассматривать как части одного слабо связанного параллельного задания. Приложения такого класса обрабатываются в проектах, созданных на платформе BOINC (SETI@Home, ClimatePrediction.net и др.).

2. Объединение ресурсов в рамках временных коллективов. Грид чаще всего ассоциируется с рекордными по необходимыми вычислительным ресурсам задачами. Простота одноуровневой организации может сделать грид обыденным средством для решения класса задач, где требуется кратное (а не на порядки) увеличение мощности по сравнению с мощностью персональных компьютеров. Для этого необходимо, чтобы один центр управления гридом был способен поддерживать деятельность нескольких небольших коллективов, в которые объединяются лица, предоставляющие ресурсы, и лица, их использующие. Такие коллективы могут образовываться динамически, и их можно рассматривать как аналоги крупных виртуальных организаций современных гридов. Диспетчер одноуровневого грида должен обеспечивать автономность подобных «виртуальных организаций» в рамках объединённой ресурсной инфраструктуры многих таких организаций.

3. Персональный грид. Вычислительные средства, которыми располагает отдельный пользователь, как правило, ограничены единственным компьютером. Накопление ресурсного парка создаёт предпосылки для преодоления этого «барьера одного компьютера», но необходима адекватная техническая и технологическая поддержка вычислительной деятельности в более сложной среде, состоящей из нескольких компьютеров. В качестве средства, обеспечивающего такую поддержку, может выступать одноуровневый грид. Владелец нескольких компьютеров может подключить их к гриду и образовать персональную виртуальную организацию, запрещая доступ к своим ресурсам всем, кроме себя самого. В результате он получает общую точку доступа ко всей совокупности компьютеров — через управляющий центр грида, который обеспечивает эффективную балансировку их загрузки. Аналогичный эффект можно получить и от кластеризации компьютеров, но подход грида исключает проблемы обслуживания кластера.

4. Предоставление дистанционного доступа к приложениям. Понятие ресурса в гриде является очень широким и не ограничивается только

системными ресурсами компьютеров (процессор, память, дисковое пространство). Ресурсом также может являться, например, устройство, подключенное к сети, а также любое приложение, которое по каким-либо причинам не может быть установлено у всех, кто хочет обрабатывать с его помощью свои данные. Путём подключения к гриду владелец компьютера, на котором установлено такое приложение, может предоставить к нему доступ и определить круг лиц, которые могут им пользоваться. Альтернативой может служить оформление приложения в виде грид-службы, но этот вариант более сложен.

Приведённые сценарии использования одноуровневой организации ресурсов свидетельствуют о наличии ситуаций, где применение такой модели даёт дополнительные возможности при решении вычислительных задач.

Цель и задачи работы

Целью диссертационной работы является разработка методов управления заданиями в одноуровневом гриде с неотчуждаемыми ресурсами, то есть используемыми совместно с их владельцами. Для достижения поставленной цели в работе решаются следующие основные задачи.

1. Исследование существующих способов диспетчеризации заданий в распределённой вычислительной среде, состоящей из некластеризованных компьютеров.
2. Разработка архитектуры системы управления инфраструктурой из некластеризованных компьютеров. Архитектура должна быть согласована с принципами и стандартами грида, что позволит создавать интероперабельные системы. Также в архитектуре должны содержаться функции, необходимые для условий неотчуждаемых компьютеров.
3. Разработка метода планирования распределения заданий, направленного на минимизацию нарушений заданного срока выполнения заданий, и оценка качества планирования на основе разработанного метода.
4. Программная реализация системы диспетчеризации для рассматриваемой формы грида.

Научная новизна

В диссертации разработан новый метод управления заданиями, направленный на минимизацию нарушения завершения заданий в заданный срок и позволяющий повысить степень загрузки компьютеров для грида с некластеризованными неотчуждаемыми ресурсами. Предлагаемый подход к распределению заданий основывается на учёте статистической информации о загрузке компьютеров. Эта информация используется для предсказания остающейся свободной доли вычислительных ресурсов в будущем.

В рамках этого подхода разработан оригинальный алгоритм планирования, использующий при принятии решения информацию об эффективной производительности компьютеров — средней величине процессорной мощности компьютера, которая остаётся невостребованной при работе владельца и может быть использована для выполнения заданий

грида. С помощью моделирования показано преимущество разработанного алгоритма по сравнению с обычно применяемыми и сформулированы предложения по эффективному использованию пула неотчуждаемых компьютеров.

Предложена архитектура системы диспетчеризации заданий, выполненная в соответствии с требованиями, предъявляемыми к программному обеспечению одноуровневого грида. В частности, в предложенной архитектуре учитывается свойство неотчуждаемости ресурсов, для чего предусмотрены средства, обеспечивающие на исполнительном компьютере в первую очередь выполнение процессов владельца, автономность ресурсов компьютера, а также динамичность среды.

Разработаны механизмы управления заданием на некластеризованных компьютерах: запуска, получения информации о загрузке компьютера (ресурсы, потреблённые заданиями грида, и остающаяся свободная мощность компьютера), ограничения количества потребляемых системных ресурсов.

Практическая значимость

На основе архитектурных решений, концепций и методов, предложенных в диссертационной работе, реализована система диспетчеризации заданий для грида с некластеризованными ресурсами. Разработанная система позволяет объединять в грид-инфраструктуры обычные компьютеры для решения важных прикладных научных и производственных задач, используя эти компьютеры в режиме разделения с их владельцами.

Практическая значимость работы подтверждается использованием разработанной системы для задач проектирования физических устройств, а также для проведения вычислительных экспериментов при моделировании физических процессов.

Апробация работы

Основные результаты работы докладывались и обсуждались на следующих конференциях и семинарах:

1. 1-я международная конференция «Распределённые вычисления и грид-технологии в науке и образовании». Доклад «Политика предоставления ресурсов в грид», Дубна, 29 июня–2 июля 2004 г.
2. Семинар МГУ им. М.В. Ломоносова «Проблемы современных информационно-вычислительных систем» под руководством д.ф.-м.н. В.А. Васенина. Доклад «Способы планирования в гриде и их реализация в грид-диспетчере», Москва, 12 апреля 2005 г.
3. Семинар группы разработчиков программного обеспечения для грид-инфраструктуры EGEE ARDA под руководством M. Lamanna. Доклад «KIAM in GT4 Evaluation Activity and Grid Research», CERN, Женева, 12 октября 2005 г.

4. 13-я Всероссийская научно-методическая конференция «Телематика-2006». Доклад «Создание прототипа центра базовых грид-сервисов нового поколения для интенсивных операций с распределёнными данными в федеральном масштабе», Санкт-Петербург, 5–8 июня 2006 г.
5. 2-я международная конференция «Распределённые вычисления и грид-технологии в науке и образовании». Доклад «Способ построения грид из некластеризованных ресурсов», Дубна, 26–30 июня 2006 г.
6. Научная конференция «Ломоносовские чтения», факультет ВМиК МГУ им. М.В. Ломоносова. Доклад «Способ построения грида из некластеризованных ресурсов», Москва, 16–24 апреля 2008 г.
7. 3-я международная конференция «Распределённые вычисления и грид-технологии в науке и образовании». Доклад «Механизмы управления разделяемыми компьютерами в гриде», Дубна, 29 июня–4 июля 2008 г.
8. 4-я международная конференция «Распределённые вычисления и грид-технологии в науке и образовании». Доклад «Применение грида с некластеризованными ресурсами для задач проектирования физических устройств», Дубна, 28 июня–3 июля 2010 г.

По материалам диссертации опубликовано 7 печатных работ [1], [2], [3], [4], [5], [6] и [7], в том числе, одна [4] — в журнале, рекомендованном ВАК для публикации основных результатов докторских и кандидатских диссертаций по вычислительной технике и информатике.

Структура и объём работы

Работа состоит из введения, пяти глав, заключения, списка литературы и двух приложений. Общий объём диссертации — 128 страниц. Список литературы содержит 60 наименований. В работе содержится 20 рисунков и одна таблица.

Содержание работы

Во **введении** обосновывается актуальность и практическая значимость диссертационной работы, рассматриваются цель и задачи исследования, а также приводится краткое содержание диссертации.

Первая глава является обзорной и содержит анализ существующих решений для организации вычислительных инфраструктур из некластеризованных компьютеров. При рассмотрении наиболее развитых средств выделены четыре подхода: первый основан на создании проектов, направленных на решение отдельных задач; второй подход состоит в применении P2P-технологий и объединении исполнительных компьютеров в одноранговые сети; к третьему подходу отнесены системы с централизованным управлением; четвёртый подход представлен частными разработками корпоративных систем. Анализ систем, реализующих каждый из представленных подходов, показал, что ни одна из них не решает всей

совокупности задач, существенных для создания системы управления одноуровневым гридом.

Так, системы, предназначенные для поддержки проектов, не предусмотрены для обработки произвольных приложений, так как для каждого нового приложения необходимо создавать свою программно-аппаратную инфраструктуру. Для такого рода систем характерно отсутствие средств запуска и управления произвольными пользовательскими приложениями, оформляемыми в виде заданий.

Слабым местом систем, реализующих подход с централизованным управлением, является планирование. В частности, ни в одной из известных систем не предлагаются решения по обеспечению выполнения задания в заданный срок. Необходимость в некоторых системах внесения изменений в код приложения или разделения его на серверную и клиентскую части также сужает область их применения. Кроме того, в таких разработках и системах, представляющих P2P-подход, не используются ставшие фактическими стандартами протоколы грида. Это приводит к тому, что подобные разработки не являются интероперабельными с другими грид-системами. То же самое можно сказать практически обо всех корпоративных решениях.

На основе проведенного анализа существующих систем в заключительной части главы предложены требования к программному обеспечению грида с некластеризованными компьютерами, выполнение которых позволит учитывать специфику таких ресурсов и организовывать на основе разрабатываемого программного обеспечения одноуровневые грид-инфраструктуры. Среди основных условий, которым должно удовлетворять программное обеспечение одноуровневого грида, выделяются:

- *запуск и управление заданиями на некластеризованных ресурсах* — приём заданий, описанных в общепринятой форме, распределение заданий по исполнительным компьютерам, получение информации о задании и отмена его выполнения;
- *стандартизованность* — соответствие общепринятым стандартам грида, обеспечивающее возможность функционирования соответствующих грид-инфраструктур как автономно, так и в составе объемлющего грида с любыми формами организации ресурсов;
- *поддержка виртуальных организаций* — обеспечение разграничения доступа к ресурсам, принадлежащим разным виртуальным организациям (динамическим объединениям пользователей);
- *интероперабельность* — предоставление информации о ресурсах поддерживаемой инфраструктуры, а также запуск заданий в стандартной для грида форме.

Специфические особенности неотчуждаемых ресурсов определяют дополнительные требования:

- *соблюдение приоритетов* — выполнение задания грида на исполнительном компьютере не должно мешать работе владельца;

- *автономность ресурсов* — сохранение за владельцем полного контроля над своим компьютером и закрепление за ним права решать, кому, когда и сколько ресурсов он готов предоставить;
- *динамичность среды* — учёт непредсказуемости количества доступных ресурсов и времени, когда эти ресурсы доступны, а также обеспечение автоматического включения компьютеров в состав ресурсной базы грида.

За исключением поддержки виртуальных организаций, все предлагаемые в диссертационной работе решения соответствуют указанным выше требованиям, реализованным в разработанной системе диспетчеризации. Требование поддержки виртуальных организаций заслуживает отдельного рассмотрения и выходит за рамки поставленных в работе задач.

Во **второй главе** предлагается архитектура системы диспетчеризации заданий для грида с некластеризованными ресурсами.

В первой части главы описываются состав, структура и функции компонентов системы, которые отвечают сформулированным в первой главе требованиям. Архитектура состоит из трёх компонентов: диспетчера, агента, устанавливаемого на исполнительном компьютере, и пользовательского интерфейса (рис. 1).

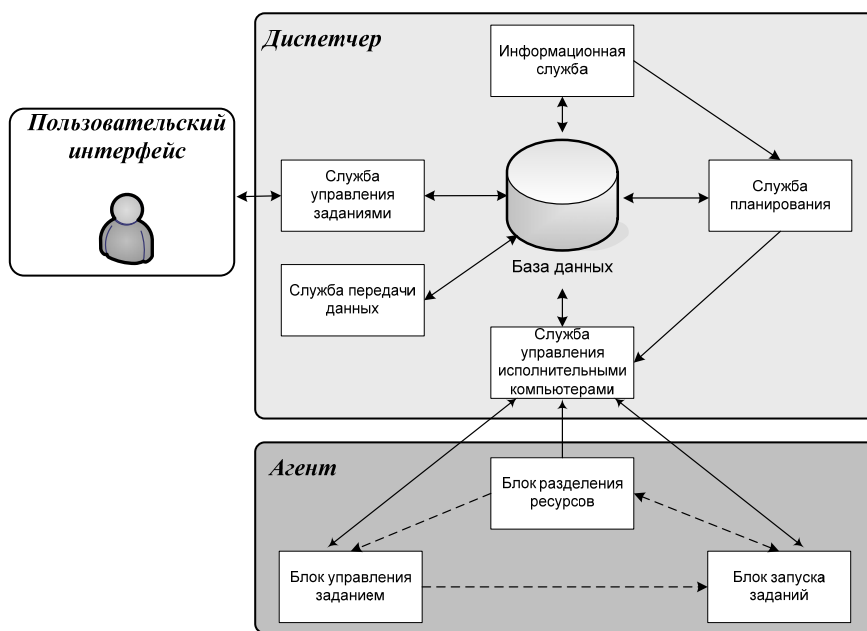


Рис. 1. Архитектура системы диспетчеризации

Диспетчер одноуровневого грида реализует интерфейсы доступа к гриду с некластеризованными ресурсами, выполняя распределение заданий между зарегистрированными в нём компьютерами и обеспечивая поддержку полного цикла обработки заданий. Агентское программное обеспечение устанавливается на исполнительных компьютерах, владельцы которых хотят

предоставить свои ресурсы в грид. Агент выполняет запуск задания и управляет им на исполнительном компьютере, осуществляя разделение ресурсов между процессами владельца и заданием грида, а также обеспечивает передачу данных и безопасность. Пользовательский интерфейс представляет собой клиентское приложение, созданное с помощью API служб диспетчера, и реализует функции управления заданиями стандартным для грида способом (ввести новое задание, получить информацию о состоянии задания, отменить выполнение задания).

Вторая часть посвящена рассмотрению диспетчера одноуровневого грида. Диспетчер представляется набором грид-служб, в котором реализованы необходимые компоненты программного обеспечения грида, а также средства взаимодействия с агентами. Как известно, грид-службы опираются на стандарты веб-служб, которые, во-первых, широко применяются при построении распределённых систем, а во-вторых, используются при разработке грид-приложений. Кроме того, реализация подсистем в виде служб позволяет получить модульную систему. В состав диспетчера входят следующие службы:

- служба взаимодействия с агентами (предоставляет интерфейс автоматической регистрации исполнительных компьютеров, заносит информацию о компьютерах в реляционную базу данных, а также принимает от агента сведения о состоянии задания, количестве потреблённых им ресурсов и загрузке компьютера);
- служба управления заданиями (получает от пользователя описание задания, заносит полученную информацию в базу данных и сообщает идентификатор задания пользователю, позволяет узнать информацию о состоянии задания и количестве потреблённых заданием ресурсов, а также удалить задание);
- служба планирования (распределяет задания по исполнительным компьютерам);
- информационная служба (осуществляет обработку непосредственно поставляемой в базу информации о загрузке ресурсов и агрегирует её для предоставления другим грид-системам);
- служба передачи данных (осуществляет доставку файлов задания на исполнительные компьютеры и во внешние хранилища).

В третьей части описывается агент, устанавливаемый на исполнительном компьютере. Основная функция агента состоит в запуске задания и управлении им на стадии выполнения. Во время управления заданием агент следит за загрузкой компьютера и принимает решение о возможности предоставления ресурсов для выполнения заданий грида. Если компьютер готов к выполнению задания грида, агент получает необходимые для задания файлы, формирует исполнительную среду и запускает его. При этом агент осуществляет разделение ресурсов между процессами владельца и заданием грида, а также следит за потреблением ресурсов заданием грида. По

запросу диспетчера агент снимает задание с выполнения. На протяжении всего времени выполнения агент посылает диспетчеру отчёты о своём состоянии и загрузке компьютера.

Заключительная часть главы посвящена пользовательскому интерфейсу, который принимает от пользователя файл с описанием задания на языке RSL, запускает задание и предоставляет возможность пользователю управлять им стандартным для грида способом. Во время выполнения задания пользователь может получить информацию о состоянии задания, количестве потреблённых заданием ресурсов, а также остановить выполнение задания.

В **третьей главе** рассматриваются методы планирования, которые могут быть применены для распределения заданий в гриде с некластеризованными ресурсами. Глава состоит из трёх частей.

В первой части даётся классификация условий планирования в зависимости от имеющейся у планировщика информации, на основе которой он принимает решение о направлении задания на тот или иной исполнительный компьютер. Так, планировщик может применять различные стратегии распределения в зависимости от полноты имеющейся информации о заданиях и компьютерах, поставленных целей, а также режима планирования.

Вторая часть посвящена сравнению алгоритмов планирования по различным условиям. Рассматриваются как случайный алгоритм и его модификации, так и алгоритмы, учитывающие при принятии решения информацию о ресурсах и заданиях. В последнем случае описываются различные способы выбора очередного задания и подходящего для его выполнения исполнительного компьютера, позволяющие повысить качество планирования.

Рассмотренные алгоритмы обладают недостатками, не позволяющими использовать их в явном виде для решения поставленной в работе задачи. Это связано с тем, что ни в одном из предложенных способов не учитывается свойство неотчуждаемости компьютеров. В такой ситуации планировщику, реализующему подобные алгоритмы, приходится принимать решение о размещении задания, основываясь лишь на характеристиках оборудования компьютеров и мгновенных показателях их загрузки, что неблагоприятно сказывается на качестве планирования и ведёт к тому, что выполнение задания может продолжаться сколь угодно долго.

В третьей части главы приводится описание предложенного автором метода планирования, направленного на минимизацию нарушения завершения заданий в заданный срок.

В связи с неотчуждаемым режимом использования ресурсов компьютера возникает ситуация, при которой доля ресурсов, получаемая заданием грида, меняется в зависимости от активности владельца компьютера. С точки зрения системы диспетчеризации такие ресурсы являются *недетерминированными*, так как невозможно точно предсказать их количество и время окончания задания, выполняющегося на неотчуждаемом компьютере. Это порождает проблему планирования процесса обработки,

одна из целей которого состоит в стремлении обеспечить завершение выполнения каждого задания в заданный предельный срок. Гарантировать это нельзя, и в качестве цели планирования в диссертации рассматривается минимизация нарушений задаваемого пользователем грида предельного времени выполнения задания.

Для планирования с таким целевым критерием наиболее важными являются характеристики, определяющие загрузку ресурсов и позволяющие её прогнозировать. Как правило, в гриде обрабатываются задания с длительным временем выполнения (несколько часов), и для планирования их распределения между компьютерами полезным выглядит использование статистических, то есть усреднённых по большим интервалам времени, характеристик компьютеров. Предлагаемый метод направлен на улучшение известных методов планирования путём построения прогноза загрузки ресурсов на основе статистических характеристик. В качестве основного показателя, характеризующего функционирование компьютера с точки зрения его работы на грид, вводится понятие *эффективной производительности*, которая определяется как средняя величина неиспользованной владельцем процессорной мощности на некотором интервале времени.

Далее в диссертационной работе даётся формальная постановка задачи планирования, а также приводится сравнение качества планирования алгоритмов *FCFS* (First Come First Serve) и его модификации *ECP-FCFS* (Effective Computer Power-FCFS). Критерием выступает число отказов по превышению предельного срока.

Задача планирования решается в следующих условиях.

- В грид поступает поток заданий.
- Алгоритм планирования выполняет распределение заданий по компьютерам в режиме on-line, то есть каждое задание распределяется независимо от других.
- Исходными данными для планирования являются время выполнения задания на эталонном компьютере, предельный срок выполнения задания и эффективная производительность компьютера.

Алгоритм FCFS.

1. Если очередь не пуста, и имеются свободные компьютеры, то выбирается случайный компьютер и на него распределяется первое задание из очереди.

2. В противном случае ожидаются события поступления задания или освобождение компьютера.

Алгоритм ECP-FCFS.

В этом алгоритме на шаге 1 меняется способ выбора исполнительного компьютера:

1.1. Выбирается первое задание из очереди.

1.2. Задание распределяется только на тот компьютер, где оно может закончиться в срок. Если таких компьютеров нет, делается попытка распределить следующее задание из очереди.

Для оценки качества планирования по предложенному алгоритму ECP-FCFS используется моделирование работы алгоритмов FCFS и ECP-FCFS на синтетических входных данных, которые описывают поток заданий и характеристики компьютеров ресурсного пула.

Компьютеры в количестве $N_c=100$ имеют эффективные производительности, генерируемые случайным образом по равномерному распределению из диапазона $[0.1, 1.0]$. Длины заданий и времена их поступления также задаются равномерным распределением. Диапазон длин заданий — $[150, 750]$, задания поступают на временном отрезке $[0, TS]$, где TS — минимально возможное время обработки потока, вычисляемое как отношение суммарной длины всех заданий к суммарной производительности компьютеров. Предельный срок выполнения заданий выбирается пропорционально их длине $D_i=\alpha_i*W_i$. Коэффициент α_i фактически определяет нижнюю границу эффективной производительности, при которой задание может выполняться в срок. Значения α_i равномерно распределены в диапазоне $[1.1, 5.0]$.

В результате моделирования обработки пакета заданий получено, что количество отказов алгоритма FCFS составляет 20.1% от общего числа заданий, в то время как количество отказов алгоритма ECP-FCFS — 3%. Улучшение, которое даёт учёт эффективной производительности, — значимое, но не слишком большое. Объясняется это тем, что компьютеры со слабой производительностью, на которых в основном и происходят отказы, вносят небольшой вклад в суммарную производительность грида. Если число таких компьютеров увеличивается, и соответственно растёт их вклад, преимущество ECP-FCFS становится весьма существенным. Это подтверждается моделированием, которое показывает, что при росте суммарной производительности пула компьютеров, в алгоритме FCFS число отказов линейно увеличивается, так как с большей вероятностью задания попадают на компьютеры с недостаточной производительностью. В то же время в алгоритме ECP-FCFS при недостаточной производительности компьютера для выполнения задания оно не распределяется на него, и число отказов остаётся постоянным.

Из-за неточности предсказания загрузки компьютера прогнозируемая производительность будет отличаться от получаемой заданием во время выполнения. Отдельная серия экспериментов была направлена на изучение влияния точности прогноза на число отказов. Относительное отклонение реально получаемой производительности H_r от прогнозируемой H задаётся величиной R , $H_r=H*(1+R)$. Значения R генерируются по нормальному распределению со средним значением 0 и стандартным отклонением σ . Величина σ является параметром исследования: при $\sigma=0$ время выполнения заданий совпадает с прогнозируемым, при больших σ оно может значительно

отклоняться от прогноза. Моделирование показывает, что использование прогноза в алгоритме ECP-FCFS уменьшает число отказов в исследованном диапазоне $0 \leq \sigma < 1$. При добавлении в пул слабых компьютеров эффект становится ещё более заметным.

Последний раздел главы посвящён повышению эффективности использования пула неотчуждаемых компьютеров. Помимо уменьшения числа отказов учёт эффективной производительности даёт дополнительный эффект: при точном прогнозе ($\sigma=0$) невозможность обработки задания в срок определяется в период его нахождения в очереди, так что не выполнимое в срок задание даже не распределяется на компьютер. Когда $\sigma \neq 0$, возможны два вида отказов: по превышению времени ожидания в очереди и по превышению времени выполнения. Наличие прогноза позволяет, во-первых, заблаговременно информировать пользователей о невозможности обработки их заданий в срок и, во-вторых, не тратить ресурсы впустую. Для оценки того, насколько продуктивно используется мощность ресурсного пула, используется степень полезной загрузки ресурсов теми заданиями, которые завершаются в срок. Моделирование работы алгоритмов показало, что для пула компьютеров с равномерным распределением производительностей полезная загрузка составляет 40%. Когда к нему добавляются слабые компьютеры, суммарная мощность возрастает, но доля полезной загрузки уменьшается до 15%.

Причина того, что задания получают отказ в обслуживании, в то время как в системе имеются незанятые компьютеры, причём в любом количестве, состоит в том, что эти компьютеры имеют недостаточно высокую производительность для выполнения заданий в срок. Более полная загрузка ресурсов может быть достигнута с помощью известного способа запуска заданий в виде пакетов подзаданий (Bag-of-Tasks). Согласно этому способу задание разбивается на множество подзаданий, каждое из которых выполняет часть вычислений. Наибольший эффект такое разбиение даёт, если подзадания являются независимыми друг от друга и могут выполняться параллельно. Тогда можно обработать всю совокупность подзаданий в срок, определённый для исходного задания, но при меньших требованиях к производительности компьютеров. Таким образом, для обработки задания удастся использовать большее число слабых компьютеров и увеличить степень полезной загрузки пула.

Четвёртая глава посвящена программной реализации системы диспетчеризации заданий для грида с некластеризованными ресурсами (SARD — StandAlone Resource Dispatcher), выполненной в соответствии с предложенной во второй главе архитектурой (рис. 1).

В первой и второй частях главы описаны состав и функции системы SARD с учётом использования стороннего программного обеспечения базового уровня. Так, в реализации SARD для управления заданиями на исполнительных компьютерах используется система Condor (рис. 2), обеспечивающая выполнение функций по обработке задания на исполнительном компьютере: распределение локальных ресурсов

компьютера, доставку всех необходимых файлов с машины диспетчера на исполнительный компьютер, а также доставку результата выполнения заданий с исполнительного компьютера на машину диспетчера.

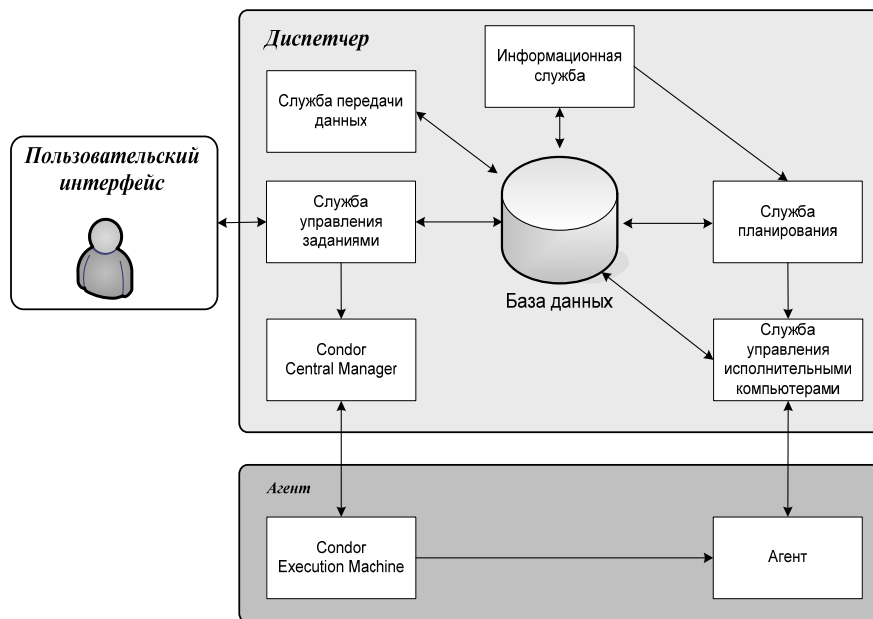


Рис. 2. Архитектура с интеграцией компонентов системы Condor

Применение в SARD программного инструментария Globus Toolkit 4 позволило построить систему, интероперабельную с существующими разработками, которые также основаны на этом инструментарии, предоставило возможность воспользоваться готовыми и отлаженными службами базового уровня, обеспечивающими, в частности, безопасность, а также поместить службы SARD внутрь контейнера Globus Toolkit 4.

Третья часть главы посвящена реализации служб, входящих в состав диспетчера системы SARD. Рассмотрены функции каждой из служб диспетчера с учётом использования стороннего программного обеспечения. Добавление системы Condor внесло изменения в процесс обработки задания — после того как планировщик определил подходящий исполнительный компьютер для задания, средствами системы Condor оно отправляется на выбранный компьютер вместе со всеми необходимыми файлами и запускается. После запуска задания на исполнительном компьютере идентификатор системного процесса сообщается агенту, который начинает следить за выполнением процессов задания и количеством потребляемых ими ресурсов.

Учитывая, что в текущей реализации SARD доставку всех необходимых заданию файлов с машины диспетчера на исполнительный компьютер и доставку результата с исполнительного компьютера обратно на машину диспетчера осуществляет система Condor, передача файлов происходит в два этапа. Сначала осуществляется доставка файлов задания и результатов вычислений между внешними хранилищами и машиной диспетчера —

внешняя доставка, а затем система Condor производит внутреннюю доставку, то есть передачу файлов между диспетчером и исполнительным компьютером. Служба передачи данных осуществляет функцию внешней доставки, для которой используется протокол GridFTP, являющийся де-факто стандартным средством передачи файлов в гриде.

В четвёртой части описывается программная реализация агента на исполнительном компьютере. Агент выполняет функции по регистрации исполнительных компьютеров в системе диспетчеризации SARD, сбору информации о потреблении ресурсов компьютера, а также отправке периодических отчётов с собранной информацией, сообщениями о функционировании самого агента и событиями, связанными с изменением статуса задания грида. В части описания агента рассмотрена его модульная архитектура, позволяющая эффективно разделить кроссплатформенные и системно-зависимые компоненты агента, облегчив тем самым создание версий агента для различных платформ, а также модернизацию агента.

Пятая часть главы посвящена рассмотрению различных сценариев работы с системой через пользовательский интерфейс, который позволяет запустить задание, отменить его выполнение, а также получить информацию о статусе выполнения задания. В реализации пользовательского интерфейса используется язык управления заданиями, совпадающий с Globus Toolkit 4.

В пятой главе содержится описание прикладных задач, решённых с помощью разработанной системы. В Институте прикладной математики им. М.В. Келдыша РАН создана действующая экспериментальная вычислительная инфраструктура из некластеризованных ресурсов. В состав инфраструктуры входит диспетчер, расположенный на выделенном компьютере под управлением операционной системы Linux, а также десять исполнительных компьютеров под управлением операционной системы Windows различных версий производительностью от 200 до 700 мегафлопс.

С помощью созданной инфраструктуры решены производственные задачи двух классов. Первый класс представляется задачами моделирования на основе метода Монте-Карло. С помощью системы SARD был произведён расчёт пространственного распределения энергии ионизирующего излучения в многокомпонентных объектах. Примерами конкретных постановок из этой области являются задачи, связанные с функционированием аппаратуры и оборудования в полях ионизирующих излучений, дефектоскопии, важной является проблема защиты этих объектов от нагрева и других поражающих факторов проникающего излучения. Особое место в данной проблематике занимает задача оценки воздействия проникающих излучений на человека.

В рассматриваемом классе задач обычно необходимо моделирование большого количества событий, что требует существенных временных затрат даже при выполнении на современных компьютерах. В то же время моделирование всех событий может быть произведено частями параллельно на нескольких компьютерах. В этой ситуации разделение «большой» задачи на несколько «небольших» подзадач и их одновременное выполнение позволяет кратно уменьшить время расчёта.

Задачи второго класса состоят в поиске оптимального значения параметров в серии вычислительных экспериментов. В рамках этого класса была решена задача расчёта движения лайнера в магнитном компрессоре, который является частью установки, разрабатываемой для исследований термоядерного синтеза. Работа компрессора основана на сжатии магнитного потока лайнером, ускоренным электродинамическими силами до скорости 1 км/с.

Одним из ключевых вопросов, стоящих перед создателями магнитного компрессора, является определение такого набора входных параметров устройства, при котором генерируемый на выходе импульс имеет оптимальные характеристики. Во время проведения вычислительного эксперимента в серии расчётов были получены результаты, позволившие подтвердить теоретические предположения, выявить оптимальные значения некоторых параметров, а также способствовали уточнению программного кода задачи. При этом использование вычислительной инфраструктуры из некластеризованных компьютеров позволило в несколько раз сократить время расчёта задачи и осуществить расчёт большего количества значений параметров.

Проведённые экспериментальные расчёты позволяют сделать выводы о том, что существует широкий класс задач вычислительного эксперимента и технических расчётов, для которых использование инфраструктуры неотчуждаемых компьютеров открывает новые возможности.

В проведённых экспериментах показано, что, во-первых, во многих задачах математической физики применяется метод Монте-Карло, который допускает параллельное выполнение одного варианта на многих компьютерах. Задачи с подобными свойствами известны и в других областях (анализа изображений, финансовой сфере, планировании, проектировании и др.). Во-вторых, можно параллельно производить расчёты для различных наборов исследуемых параметров.

В **заключении** перечисляются основные результаты работы.

Основные результаты

1. Разработан новый подход, позволяющий создавать вычислительные инфраструктуры из компьютеров, которые используются в режиме разделения с их владельцами. Предложены сценарии применения таких инфраструктур для массовой обработки вычислительных заданий.
2. Предложена архитектура системы управления инфраструктурой из некластеризованных компьютеров. Архитектура согласована с принципами и стандартами грида, в ней введены функции, необходимые для условий неотчуждаемых компьютеров. Предложенная архитектура допускает использование некластеризованных ресурсов в составе объёмлющих грид-инфраструктур с любой формой организации ресурсов.

3. Разработан новый метод планирования распределения заданий, направленный на обеспечение завершения заданий в заданный срок. Путём моделирования показано, что использование прогноза эффективной производительности исполнительных компьютеров существенно повышает качество планирования и позволяет увеличить степень полезной загрузки пула неотчуждаемых компьютеров.
4. На основе предложенных решений выполнена программная реализация системы диспетчеризации, создана экспериментальная инфраструктура, на которой проведены расчёты производственных задач.

Список публикаций по теме диссертации

- [1]. П.С. Березовский, В.Н. Коваленко. Политика предоставления ресурсов в грид // Распределённые вычисления и Грид-технологии в науке и образовании: Труды международной конференции. Дубна: ОИЯИ, 2004. С. 36–41.
- [2]. Березовский П.С., Коваленко В.Н. Способ построения грид из некластеризованных ресурсов // Распределённые вычисления и Грид-технологии в науке и образовании: Труды 2-й международной конференции. Дубна: ОИЯИ, 2006. С. 216–226.
- [3]. П.С. Березовский, В.Н. Коваленко. Состав и функции системы диспетчеризации заданий в гриде с некластеризованными ресурсами // Препринт № 67. Москва: ИПМ им. М.В.Келдыша РАН, 2007. 29 с.
- [4]. П.С. Березовский, В.Н. Коваленко. Планирование в гриде с разделяемыми ресурсами на основе статистических данных // Программные продукты и системы. 2009. № 1(85). С. 3–6.
- [5]. П.С. Березовский. Реализация системы диспетчеризации заданий SARD в одноуровневом гриде // Препринт № 49. Москва: ИПМ им. М.В.Келдыша РАН, 2010. 32 с.
- [6]. П.С. Березовский, В.Н. Емельянов, В.Н. Коваленко, Э.С. Луховицкая. Механизмы управления разделяемыми компьютерами в гриде // Распределённые вычисления и Грид-технологии в науке и образовании: Труды 3-й международной конференции. Дубна: ОИЯИ, 2008. С. 303–306.
- [7]. П.С. Березовский, А.С. Родин. Проведение оптимизационных расчётов для магнитного компрессора с использованием грида персональных компьютеров, управляемых системой SARD // Препринт № 7. Москва: ИПМ им. М.В.Келдыша РАН, 2011. 31 с.