УДК 681.3.06

А.Е. Дорошенко, О.В. Алистратов, Ю.М. Тырчак, А.П. Розенблат, К.А. Рухлис

СИСТЕМЫ GRID-ВЫЧИСЛЕНИЙ — ПЕРСПЕКТИВА ДЛЯ НАУЧНЫХ ИССЛЕДОВАНИЙ

Анализируется развитие Grid-систем, насчитывающих уже три поколения: от систем первого поколения, предвестников современных Grid, через системы второго поколения с программным обеспечением промежуточного уровня для поддержки крупномасштабных данных и вычислений — к системам третьего поколения, где центром внимания становится сервисно-ориентированный подход и распределенное глобальное сотрудничество. Обсуждается соотношение между Grid и Web и рассматривается перспектива использования конвергенции Web- и Grid-технологий в научных исследованиях.

Введение

В последнее десятилетие произошли значительные изменения в способах восприятия и использования вычислительных ресурсов и услуг. Если раньше было нормально удовлетворять вычислительные потребности через локальные вычислительные платформы и инфраструктуры ограниченного характера, т.е. персональные компьютеры и локальные сети, то сегодня ситуация меняется. Это связано, среди прочих факторов, с увеличением количества пользователей компьютеров и сетевых компонентов, появлением более быстрых и развитых аппаратных средств и все более сложного программного обеспечения. Следствием таких изменений стала возможность эффективного использования широко распределенных ресурсов в широком диапазоне областей применения, в том числе и научных исследованиях. Все более мощные и гибкие вычислительные технологии порождают новые возможности сегодняшней науки в моделировании, анализе данных и формах научного сотрудничества. Однако, несмотря на то, что показатели производительности компьютеров, средств хранения данных и связи продолжают увеличиваться экспоненциально, решающих успехов в использовании потенциала этих вычислительных ресурсов достичь до сих пор не удавалось в силу отставания средств комплексирования и управления сложными вычислительными комплексами, особенно глобального масштаба.

С тех пор, как стало возможным объединять ЭВМ в компьютерные сети, проблемы проектирования и развертывания распределенных компьютерных систем исследуются уже в течение многих лет. Все большее количество исследований выполняются в области распределенных широкомасштабных (глобальных) вычислений. Эти группы разрабатывают промежуточное программное обеспечение (middleware), библиотеки и инструментальные средства, которые позволяют совместно использовать территориально распределенные, но объединенные ресурсы как единую мощную платформу для выполнения параллельных и распределенных приложений. Такой подход к вычислениям ранее был известен под несколькими названиями, такими, как метакомпьютинг [1], масштабируемые или глобальные вычисления, в последнее время — вычисления Grid [2].

В недавней работе одного из пионеров этого направления [3] понятие Grid-системы определено как "гибкое, безопасное, скоординированное совместное использование ресурсов динамическими образованиями из личностей, учреждений и ресурсов". Оно подчеркивает значение аспектов информации, существенных для обнаружения ресурса и его интероперабельности в составе ансамблей с другими ресурсами. Однако жизнь не стоит на месте, и новые проекты по разработке Grid-систем [4] уже акцентируют на продолжение этой линии исследований

путем перехода от информации к знаниям. Это связано также с эволюцией Web-технологий и стандартов, таких, как XML для поддержки обменов между компьютерами, и RDF для представления метаданных.

Почему же стал возможным и необходимым такой резкий рывок в осознании нового уровня использования компьютерной техники и почему это так важно для научного мира? В [4] приведены следующие сравнительные данные. Персональный компьютер в 2001-м году имел примерно такую же скорость, как суперкомпьютер в 1990-м, а десяток лет назад биологи считали за счастье иметь возможность промоделировать на компьютере одну молекулярную структуру. Теперь персональные компьютеры оснащены сотнями Гбайт (1Гб = 10^9 байт) долговременной памяти хранения — столько, сколько имел целый суперкомпьютерный центр в 1990-м, и это дает возможность вычислять структуры сложных ансамблей макромолекул и анализировать тысячи вариантов лекарственных препаратов. К 2006-му году ожидается, что проекты по физике высоких энергий, в том числе Large Hadron Collider (LHC) Европейского центра ядерных исследований (CERN), произведут десятки петабайт (1ПБ = 10^{15} байт) данных в год. Современные широкополосные сети теперь работают со скоростью 155 Мбс (мегабит/с), что на три порядка превышает скорость в 56 кбс (килобит/с), с которой соединялись американские суперкомпьютерные центры в 1985-м году [5]. Но чтобы работать с зарубежными коллегами на наборах данных, объем которых измеряется петабайтами, ученым теперь требуются скорости в десятки гигабит в секунду (1Гбс = 10^9 бит/с).

То, что теперь понимают под термином Grid, представляет собой инфраструктуру, построенную на основе Интернета и Всемирной паутины (World Wide Web), которая обеспечивает масштабируемые, безопасные и быстродействующие механизмы для обнаружения и доступа к удаленным вычислительным и информационным ресурсам. Технологии Grid могут дать возможность ученым разделять ресурсы для географически распределенных групп в беспрецедентном масштабе и такими способами, которые ранее были просто невозможны [2 — 4].

В данной статье анализируются научные и технологические достижения, сделавшие возможным построение Grid-систем, а также рассмотрены основные перспективы использования конвергенции Web- и Grid-технологий для решения сложных научно-технических задач и автоматизации научных исследований.

1. Качественный скачок в коммуникационных технологиях

Для оценки технологического изменения полезной метрикой обычно считается средний период времени, в течение которого удваиваются основные функциональные показатели изделий (производительность, емкость и т.п.) или, что более или менее эквивалентно, вдвое уменьшается их цена. Для памяти хранения, скорости сетей и вычислительной мощности эти периоды составляют около 12, 9, и 18 месяцев соответственно. Различное значение констант, связанных с этими тремя экспонентами, имеет принципиальное значение.

Ежегодное удвоение емкости устройств хранения данных, измеряемое количеством бит на единицу площади, уже снизило стоимость терабайта (1ТБ = 10^{12} байтов) дисковой памяти до величины ниже $10 000. Ожидается, что тенденция продолжится, и разработчики в области экспериментальной физики и высокоточного моделирования с большой разрешающей способностью планируют разработку архивов данных объемом в петабайты [4]. Такие большие объемы данных требуют соответствующих более мощных средств для анализа. Однако, несмотря на серьезные продвижения в индустрии микропроцессорной техники, темпы роста производительности компьютеров отстают от таковых для средств

хранения информации. Удваиваясь "только" каждые 18 месяцев, мощность компьютеров потребует пять лет для своего увеличения на порядок.

Вместе с тем разительные изменения произошли за последнее время в коммуникационных технологиях и организации компьютерных сетей. Благодаря таким нововведениям, как добавление примесей при производстве оптоэлектронных устройств, производительность широкомасштабных сетей удваивается каждые девять месяцев, а каждые пять лет это дает увеличение на два порядка. Так, сеть NSFnet, которая соединяет суперкомпьютерные центры Национального научного фонда США, наглядно иллюстрирует эту тенденцию. Если в 1985-м году магистральные каналы NSFnet использовались с беспрецедентной тогда скоростью 56 кбс [5], то в 2002-м они работали в сети TeraGrid (http://www.teragrid.org/) уже на скорости 40 Гбс, т.е. увеличение произошло на шесть порядков за 17 лет [4].

Таким образом, удвоение производительности сети по отношению к росту мощности компьютеров каждые 18 месяцев означает, что концентрация вычислительных ресурсов в одном месте для крупномасштабного анализа становится нецелесообразной. Отсюда вытекает также, что если такое соотношение темпов сохранится, то очень скоро связь станет почти бесплатной и практически неограниченной. Чтобы использовать это обилие пропускной способности, нужно находить новые способы работы сетей с интенсивной связью: объединение вычислительных ресурсов в пулы, обрабатывающие потоки большого количества данных, повсеместное внедрение датчиков и автоматизация обработки сенсорной информации в быту и на производстве, а также внедрение (колаборативных) сред совместной работы, устраняющих необходимость в дорогостоящих командировках.

Осуществление этих шагов требует универсальных механизмов для таких критических задач, как создание и управление сервисами на удаленных компьютерах, поддержка подписки на распределенные ресурсы, передача больших наборов данных на высоких скоростях, формирование больших распределенных виртуальных сообществ и поддержка информации о существовании, состоянии и политике использования их ресурсов. Сегодняшний Интернет и Web-технологии не рассматривают таких возможностей. Обеспечение инфраструктуры и инструментов для прямого крупномасштабного и безопасного использования ресурсов и явилось первейшей задачей Grid. В последующих разделах анализируются основные этапы становления и развития концепций и средств Grid-систем и технологий.

2. Первое поколение Grid

Ранние попытки по созданию Grid были предприняты как проекты по связи суперкомпьютеров между собой; в то время этот подход был известен как метакомпьютинг (metacomputing). Этот термин был употреблен в проекте CASA, одном из нескольких проектов системы гигабитной связи в США [1]. К середине 1990-х отмечают появление раннего метакомпьютинга или Grid-сред. Как правило, цель ранних проектов метакомпьютинга заключалась в обеспечении вычислительными ресурсами некоторого набора приложений, требующих высокой производительности. Двумя показательными проектами этого типа были FAFNER [6] и I-WAY [7].

Они во многом различались, но для обеспечения эффективной работы должны были решить ряд аналогичных проблем, таких, как связь, управление ресурсами и манипулирование удаленными данными. Свои решения предоставлять ресурсы метакомпьютинга они попытались выполнить с противоположных концов вычислительного спектра. Если FAFNER мог вовлекать в работу любую рабочую станцию с объемом памяти не менее 4 Мб, то I-WAY предназначался для объединения ресурсов больших суперкомпьютерных центров США.

Проект FAFNER был нацелен на решение задачи разложения на множители для алгоритма шифрования с публичными ключами по методу RSA, широко используемому, например, в протоколе безопасных соединений SSL (Secure Sockets Layer). Разложение на множители в вычислительном отношении занимает очень много времени. По этой причине были разработаны параллельные алгоритмы разложения на множители, выполнение которых можно было распределить. Алгоритмы оказались тривиально параллельными и не требовали никакой связи между отдельными ветвями после начальной установки, которая позволила многим исполнителям выполнять некоторую небольшую часть большой задачи разложения на множители. Так был создан проект по разложению на множители через Web, известный как FAFNER (Factoring via Network-Enabled Recursion).

В проекте использовался новый метод, под названием NFS (Number Field Sieve), разложения на множители на основе вычислительных Web-серверов. Исполнитель на основе Web-форм вызывал на сервере сценарии CGI (стандартного интерфейса обмена данных), написанные на языке Perl. Исполнители, основываясь на некотором наборе Web-страниц, могли обращаться к широкому набору сервисов для шагов просеивания при разложении на множители, а именно: распространению программного обеспечения NFS, документации проекта, анонимной регистрации исполнителя, распространению просеивающих заданий и др. Сценарии CGI поддерживали также администрирование кластера компьютеров, направляя активность отдельных просеивающих рабочих станций на ночное время, чтобы уменьшить пересечение с работой их владельцев. Исполнители загружали и монтировали просеивающего программного демона, который становился их Web-клиентом, использовавшим протокол HTTP для получения заданий и возвращения результатов через сценарий CGI на Web-сервере.

Чтобы сделать этот подход успешным, важны три фактора:

- реализация программы NFS позволяла рабочим станциям с 4 Мб памяти выполнить полезную работу, используя малые границы и малое решето;
- поддержка анонимной регистрации; исполнители могли добровольно давать свои ресурсы аппаратных средств, объявляясь только администратору локальной сети;
- формирование иерархической сети Web-серверов из сайтов, задействованных для локального выполнения пакета сценария CGI, с целью минимизации критической потребности в администрировании сети, что позволяло просеивать данные круглосуточно с наименьшим вмешательством оператора.

Проект FAFNER был отмечен наградой в конкурсе TeraFlop на конференции по суперЭВМ 1995 г., что позволило появиться множеству других проектов, основанных на метакомпьютинге.

Проект I-WAY, напротив, был экспериментальной высокоэффективной сетью, связывающей множество высокопроизводительных компьютеров и развитых сред визуализации. Проект начался в начале 1995 года с идеи не строить новую сеть, а интегрировать существующие сети высокой пропускной способности. Виртуальные среды, наборы данных и компьютеры, постоянно находящиеся в семнадцати различных городах США, были соединены десятью сетями с переменной пропускной способностью. При этом использовались различные технологии маршрутизации и коммутации. Сеть была основана на технологии асинхронной передачи данных ATM, которая в то время находилась в стадии становления, и поддерживала протоколы TCP/IP над ATM и непосредственно ATM-ориентированные протоколы.

Для стандартизации интерфейсов программирования и управления ключевые сайты установили I-POP-серверы ("пункты присутствия"), которые действовали как шлюзы стыковки к I-WAY.

Серверы I-WAY были рабочими станциями под UNIX, сконфигурированными для выполнения среды стандартного программного обеспечения I-Soft. На I-Soft возлагалось решение задач, связанных с неоднородностью, масштабируемостью, производительностью и безопасностью. В проекте I-WAY был также разработан планировщик ресурсов (вычислительный брокер ресурсов CRB), который состоял из протоколов CRB-пользователь и CRB-планировщик. Фактическое выполнение брокера CRB было распределено между центральным планировщиком и множеством локальных демонов-планировщиков, по одному на I-POP-сервер. Центральный планировщик вел очереди задач и таблицы, представляющие состояние локальных машин, помещая задания на обработку и информацию о состоянии вычислений в распределенную файловую систему AFS, которая давала возможность хост-машинам совместно использовать ресурсы локальных и глобальных сетей.

В I-POP-серверах защита обеспечивалась с использованием клиента telnet, модифицированного для использования аутентификации и шифрования Kerberos. Кроме того, CRB играли роль агента, исполняющего последовательное удостоверение подлинности доступа к ресурсам I-WAY от имени пользователя. Для поддержки инструментальных средств библиотека обменов Nexus поддерживала автоматические конфигурационные механизмы, которые давали возможность выбрать соответствующую конфигурацию в зависимости от используемой технологии, например связь через TCP/IP или ATM. Библиотека MPICH (переносимая реализация стандарта MPI) была также расширена для использования Nexus. Проект I-WAY определял несколько типов приложений: вычисления на суперЭВМ; доступ к удаленным ресурсам; виртуальный мир, видео и графические интерфейсы.

Таким образом, FAFNER и I-WAY попытались создать метакомпьютерную среду для интеграции ресурсов, но с разных сторон. FAFNER стал широкомасштабной вездесущей (ubiquitous) системой, которая могла работать на любой платформе с Web-сервером. Как правило, его клиентуру составляли обычные терминальные компьютеры, в то время как I-WAY объединил ресурсы в крупных центрах вычислений на суперЭВМ. Два проекта также различались типами приложений, которые могли использовать их среды. FAFNER был приспособлен к конкретному приложению разложения на множители, которое само по себе было тривиально параллельным и не зависело от быстрых коммуникаций. I-WAY был разработан для другого набора приложений различной производительности, которые обычно нуждались в быстрых обменах и мощных ресурсах. Однако оба проекта испытывали недостаток масштабируемости. Они были весьма новаторскими и успешными, и каждый проект был в авангарде метакомпьютинга, прокладывая путь ко многим последующим проектам Grid второго поколения. FAFNER стал предвестником проектов, подобных SETI@home [8] и Distributed.Net [9], а I-WAY — Globus [10] и Legion [11].

3. Второе поколение Grid

Акцент ранних работ по Grid-вычислениям был, в частности, вызван потребностью связать ряд американских национальных центров суперЭВМ, и проект I-WAY успешно достиг этой цели. Но сегодня инфраструктура Grid может объединять вместе не только специализированные центры суперЭВМ, она постепенно превращается в повсеместную распределенную инфраструктуру глобального масштаба, которая вместе с технологиями сетей высокой пропускной способности и принятием стандартов может поддерживать различные приложения, требующие крупномасштабных вычислений и данных. Такая перспектива для Grid сформулирована в [12] и расценивается как второе поколение этих систем.

Системы второго поколения сосредоточились на трех главных проблемах.

- *Неоднородность.* Grid включает множество ресурсов, которые являются разнородными по своей природе, и может пересекать многие административные границы даже в глобальном масштабе.
- *Масштабируемость.* Размер Grid может возрастать до миллионов единиц ресурсов. Это порождает проблему потенциального уменьшения ускорения вычислений по мере увеличения размера Grid. Поэтому приложения, которые требуют большого количества географически распределенных ресурсов, должны быть разработаны с учетом устойчивости к задержкам и использования местоположения ресурсов. Кроме того, поскольку увеличение масштаба также предполагает пересечение большого количества административных границ, есть потребность в решении проблемы доверия и аутентификации (удостоверения подлинности). Приложения большого масштаба могут создаваться как композиции других приложений, что увеличивает интеллектуальную сложность систем.
- *Адаптируемость.* В Grid неисправность ресурса есть правило, а не исключение, поскольку вероятность отказа при таком количестве ресурсов очень высока. Поэтому администраторы ресурсов или сами приложения должны динамически приспосабливать свое поведение так, чтобы они могли получать максимальную производительность на используемых ими ресурсах и сервисах.

Программное обеспечение (ПО) промежуточного уровня (middleware) обычно рассматривают как ПО, расслоенное между операционной системой и приложениями и предоставляющее ряд сервисов для корректного и эффективного их функционирования. В последнее время его стали воспринимать как *способ интегрирования приложений,* работающих в распределенных разнородных средах. В Grid промежуточное ПО используется, чтобы скрыть разнородность платформ и предоставить потребителям и приложениям однородную и монолитную (бесшовную — seamless) среду с набором стандартизированных интерфейсов и сервисов.

Принятие и использование стандартов также является ключевым моментом в борьбе с неоднородностью. Стандарты и системные интерфейсы API позволяют переносить сервисы и приложения на множество компьютерных систем, использующихся в среде Grid. Очевидно, что если Grid состоит из n компонентов, то лучше принять n соглашений о приведении форматов обмена к стандартному, чем иметь n^2 преобразователей каждого формата в каждый.

Далее в этом разделе рассматриваются общие требования второго поколения Grid, а также его базовые технологии, брокеры ресурсов и планировщики, полностью интегрированные системы и одноранговые (peer-to-peer) системы.

3.1. Необходимые условия для инфраструктуры данных и вычислений. Инфраструктура данных может состоять из всевозможных ресурсов с сетевыми связями от компьютеров и запоминающих устройств большой емкости до баз данных и специальных научных установок (инструментов). Главными проектными характеристиками, требуемыми от данных и вычислительной структуры Grid, являются следующие.

- *Административная иерархия.* Это способ распределения среды Grid, который потенциально способен обеспечить глобальный масштаб сети. Административная иерархия определяет, в частности, как распространяется административная информация по сети.
- *Сервисы связи.* Инфраструктура связи должна поддерживать протоколы для транспортировки массивов данных и потоковых данных, поддержки групповых обменов и использования распределенных объектов. Сетевые сервисы также обеспечивают Grid важными параметрами качества обслуживания QoS, такими, как задержка,

пропускная способность, надежность, отказоустойчивость, и др.

• *Информационные сервисы.* Grid представляет собой динамическую среду, в которой место размещения и тип доступных сервисов постоянно изменяются. Информационные сервисы Grid (регистрация и каталогизация) обеспечивают механизмы для регистрации и получения текущей информации относительно структуры, ресурсов, сервисов, состояния и свойств среды.

• *Сервисы именования.* В Grid имена используются для различения большого многообразия объектов, таких, как компьютеры, сервисы или данные. Сервис именования обеспечивает универсальное пространство имен во всей распределенной среде. Типичным примером сервиса именования является DNS, обеспечиваемый международным стандартом X.500 (схема Internet).

• *Распределенные файловые системы и кэширование.* Распределенные приложения часто требуют доступа к файлам, находящимся на разных серверах, поэтому распределенная файловая система является ключевым компонентом, она может снабжать универсальное глобальное пространство имен, поддерживает множество протоколов ввода-вывода, не требует много или вообще ничего для модификации программы и обеспечивает средства для оптимизации производительности, такие как кэши.

• *Защита и авторизация.* Любая распределенная система включает все четыре аспекта безопасности: конфиденциальность, целостность, аутентификация и отчетность (accountability). Безопасность в пределах Grid — это комплексная проблема, требующая от различных ресурсов автономного управления и взаимодействия между собой таким способом, который не влияет на использование ресурсов.

• *Состояние системы и отказоустойчивость.* Для обеспечения надежной и устойчивой среды важно наличие средств управления ресурсами и приложениями. Для этого должны быть развернуты инструментальные средства мониторинга ресурсов и приложений.

• *Управление ресурсами и планирование.* Администрирование процессорным временем, памятью, сетью, памятью хранения и другими компонентами в Grid является важной частью эффективного планирования приложений. С пользовательской точки зрения управление ресурсами и планирование приложений должны быть прозрачными, а их взаимодействие с управлением — сводиться к разрешению на выполнение приложений.

• *Пользовательский и административный графические интерфейсы.* Интерфейсы сервисов и доступных ресурсов должны быть интуитивными и легкими в использовании, а также разнородными по природе. Как правило, пользовательский и административный доступы к приложениям Grid основаны на Web-сервисах.

3.2. Базовые технологии для Grid второго поколения. В настоящее время имеется большое количество проектов, связанных с Grid и имеющих дело с такими категориями, как инфраструктура, сервисы, сотрудничество, специальные приложения и порталы домена. К наиболее значительным относится проект Globus [10].

Globus обеспечивает программную инфраструктуру, которая дает возможность приложениям работать с распределенными разнородными вычислительными ресурсами как с единой виртуальной машиной. Проект Globus — это исследовательская программа в США, ставящая своей целью построение вычислительных сетей Grid. Вычислительная сеть в этом контексте является программно-аппаратной инфраструктурой, которая обеспечивает надежный, непротиворечивый и проникающий (pervasive) доступ к предельным вычислительным возможностям систем, несмотря на географическое размещение ресурсов и их потребителей. Центральным элементом системы Globus является инструментарий Globus Toolkit, который определяет

основные сервисы и возможности, требующиеся для создания Grid. Инструментарий состоит из набора компонентов, реализующих базовые сервисы, такие, как защита, размещение ресурса, управление ресурсами и связь.

Для вычислительных сетей Grid необходимо поддерживать широкое многообразие парадигм программирования и приложений. Следовательно, вместо обеспечения универсальной модели программирования, такой как объектно-ориентированная модель, инструментарий Globus Toolkit обеспечивает набор сервисов, которые разработчики инструментальных средств или приложений могут использовать для их специфических нужд. Он создан как слоистая архитектура, в которой сервисы верхнего глобального уровня смонтированы на уже существующих низкоуровневых локальных сервисах ядра. Инструментарий является модульным и приложение может использовать функции Globus без библиотеки обменов. Инструментарий Globus состоит из следующих функций (точный набор зависит от версии):

- основанный на протоколе HTTP администратор распределения ресурсов GRAM — используется для распределения вычислительных ресурсов, контроля и управления вычислением на этих ресурсах;
- расширенная версия протокола передачи файлов GridFTP — применяется для доступа к данным; расширения включают использование протоколов защиты уровня связности, частичного доступа к файлу и управления параллелизмом для высокоскоростных передач;
- аутентификация и связанные с ней сервисы безопасности GSI (Grid Security Infrastructure);
- распределенный доступ к информации о структуре и состоянии, который основан на протоколе облегченного доступа к каталогам LDAP (Lightweight Directory Access Protocol); этот сервис используется для определения стандартного протокола информации о ресурсе и связанной с ним информационной модели;
- удаленный доступ к данным через последовательный и параллельный интерфейсы GASS (Global Access to Secondary Storage), включая интерфейс к GridFTP;
- построение, кэширование и размещение выполнимых программ GEM (Globus Executable Management);
- резервирование и распределение ресурсов GARA (Globus Advanced Reservation and Allocation).

Globus прошел в своем развитии этапы своего первого воплощения в I-WAY, первой версии GT1 и текущей версии GT2. Протоколы и сервисы Globus изменялись по мере его развития. Главный акцент перешел от поддержки высокопроизводительных приложений к более охватывающим (pervasive) сервисам, которые могут поддерживать виртуальные организации. Эволюция Globus продолжается введением Открытой архитектуры сервисов Grid (OGSA) [13].

3.3. Брокеры ресурсов и планировщики Grid. В этом разделе различаются следующие виды систем.

3.3.1. Пакетные и планирующие системы. Имеется несколько известных систем, где основное внимание уделено планированию ресурсов и пакетному выполнению заданий. Следует заметить, что все пакетные системы начали свое существование как системы управления заданиями или задачами на локально распределенных вычислительных платформах [14]. К ним относятся следующие.

- Condor [15] — это пакет программ для выполнения пакетных заданий на платформах UNIX, особенно когда они бывают незагруженными. Главными особенностями Condor являются автоматическое нахождение ресурсов, распределение работ, установка контрольных точек и миграция процессов. Эти характеристики реализованы без модификации ядра UNIX. Однако пользователю необходимо подключать библиотеки Condor к исходному тексту. Condor контролирует все

действия с участием вычислительных ресурсов, к которым принадлежат все доступные Condor компьютеры. Пул ресурсов является динамическим — компьютеры попадают в него, как только перестают использоваться, и покидают, когда получают задания.

• Переносимая пакетная система PBS (Portable Batch System) [16] — это пакетная организация очереди заданий и управления рабочей нагрузкой, первоначально разработанная для NASA. Работает на ряде платформ UNIX от кластеров на суперЭВМ. Планировщик заданий PBS позволяет локальным узлам устанавливать их собственную дисциплину планирования во времени и пространстве. PBS адаптирован к различным вариантам политики администрирования и обеспечивает расширяемую аутентификацию и модель защиты. PBS имеет графический интерфейс пользователя при регистрации заданий, отслеживании их выполнения и для других административных функций.

• Платформа Sun Grid Engine (SGE) [17] основана на программном обеспечении, разработанном фирмой Genias, известном как Codine/GRM. В SGE задания находятся в зоне ожидания, а очереди на серверах обеспечивают сервисы для заданий. Потребитель вводит задание в SGE и объявляет профиль необходимых требований для его выполнения. SGE определяет заданию соответствующую очередь и распределяет его либо с высшим приоритетом, либо с самым длинным временем ожидания, пробуя запускать новые задания на наиболее соответствующей или наименее загруженной очереди.

• Средство распределения нагрузки LSF (Load Sharing Facility) — это коммерческая система от Platform Computing Corp [18]. LSF возникла из системы Utopia, разработанной в Университете Торонто, и в настоящее время является наиболее широко используемой коммерческой системой для управления выполнением заданий. LSF включает распределенный механизм разделения нагрузки и программное обеспечение для организации очередей заданий, которое управляет, контролирует и анализирует ресурсы и рабочие нагрузки на сети разнородных компьютеров, имея при этом возможности обеспечения отказоустойчивости.

3.3.2. Брокеры памяти хранения. Брокер памяти хранения SRB (Storage Resource Broker) [19] был разработан для обеспечения универсального доступа к распределенной памяти хранения запоминающих устройств. SRB поддерживает репликацию файлов, и это может происходить как в автономном (offline), так и оперативном (on-the-fly) режимах. Взаимодействие с SRB осуществляется через графический интерфейс пользователя. Серверы SRB могут быть объединены. Каждым SRB управляет администратор с полномочиями создавать группы пользователей. Главная особенность разработки состоит в поддержке метаданных, связанных с распределенной файловой системой, таких, как место нахождения, размер и информация о дате создания, а также метаданных уровня приложений (или предметно-зависимых), специфичных для конкретного содержания, которое не может быть обобщено для всех наборов данных. В отличие от традиционных сетевых файловых систем SRB является привлекательным для приложений Grid, которые работают с большими объемами данных, превышающими вместимость индивидуальных запоминающих устройств, и потому используют метаданные со всеми преимуществами репликации файлов.

3.3.3. Брокеры ресурсов Nimrod/G и GRACE. Nimrod/G — это брокер для Grid, который выполняет управление ресурсами и планирование независимых приложений [20]. Он состоит из четырех компонентов: генератора независимых заданий, планировщика, диспетчера и агента ресурсов. Генератор независимых заданий позволяет подключать определяемые пользователем планировщики, настраиваемые приложения или решатели задач в качестве компонентов приложений по умолчанию. Диспетчер использует

Globus для развертывания компонентов Nimrod/G на удаленных ресурсах и управления выполнением назначенных заданий. Планировщики Nimrod/G имеют способность предоставлять ресурсы Grid и сервисы в зависимости от их возможностей, стоимости и доступности. Планировщики обеспечивают обнаружение ресурсов, отбор, планирование и выполнение заданий пользователей на удаленных ресурсах. Пользователи могут устанавливать конечный срок, к которому следует получить результаты, и брокер Nimrod/G пробует находить наилучшие ресурсы, доступные в Grid, и использовать их для удовлетворения пользовательских требований и минимизации затрат на выполнение задания.

Nimrod/G поддерживает определяемый пользователем конечный срок выполнения и бюджетные ограничения для планирования оптимизаций и управляет спросом и предложением ресурсов в Grid, используя набор сервисов торговли ресурсами GRACE (Grid Architecture for Computational Economy) [21]. Имеются четыре алгоритма планирования Nimrod/G [22]:

- оптимизация стоимости использует самые дешевые ресурсы, гарантирующие выполнение требования крайнего срока при наименьшей стоимости;
- оптимизация времени реализует все возможные ресурсы для скорейшего параллельного выполнения задания;
- оптимизация время — стоимость подобна стоимостной оптимизации, но при множестве ресурсов одинаковой стоимости применяется стратегия оптимизации времени;
- консервативная по времени стратегия подобна оптимизации времени, но гарантирует каждому необработанному заданию минимум расходов на одно задание.

Брокер Nimrod/G с этими стратегиями планирования использовался в решении крупномасштабных вычислительных приложений, таких, как моделирование калибровки камеры ионизации [20] и молекулярного моделирования при создания лекарств [23].

3.4. Порталы Grid. Web-портал дает возможность ученым и исследователям иметь доступ к информационным ресурсам, специфическим для их предметной области, с помощью Web-интерфейса. В отличие от обычных тематических Web-порталов портал Grid может также обеспечивать доступ к вычислительным ресурсам Grid (например, подтверждать подлинность пользователей, давать им разрешение на доступ к удаленным ресурсам, помогать принимать решения относительно планирования заданий, получать и обрабатывать информацию о ресурсах, хранящуюся в удаленных базах данных). Портал Grid может также быть индивидуализирован при помощи профилей, которые создаются и сохраняются для каждого пользователя портала. Эти свойства, как и другие, превращают порталы Grid в адекватные средства для пользователей приложений Grid при обращении к ресурсам Grid.

Портал HotPage [24] разработан для одноточечного входа в компьютерные ресурсы и упрощения доступа к распределенным ресурсам, которые представляются и как части интегральной системы Grid и как индивидуальные компьютеры.

Два ключевых сервиса, обеспечиваемых HotPage, — это информация и доступ к ресурсам, а также администрирование. Информационные сервисы разработаны с целью увеличения эффективности использования ресурсов посредством: документации пользователя и навигации; статьи текущих новостей; информации об обучении и консультациях; данных относительно платформ и приложений; информации о ресурсах, такой как регистрация пользователей и данные расчетов.

Интерактивный Web-сервис HotPage также предлагает безопасные транзакции для доступа к ресурсам и позволяет пользователю исполнить команды выполнения, трансляции и

запуска программ. Другим ключевым сервисом является информирование о статусе ресурсов и поддержка простых механизмов назначения заданий ресурсам. Информация о статусе включает: загрузку (процент использования) центрального процессора; карту процессорного узла; резюме использования очереди; текущую информацию об очереди для всех участвующих платформ.

Другим примером портала Grid является GridPort [25] — переналаживаемый портальный инструментарий, который использует инфраструктуру HotPage. Двумя ключевыми компонентами GridPort являются сервисы Web-портала и прикладные API. Модуль Web-портала работает на Web-сервере и обеспечивает безопасное подключение к Grid. Прикладные API обеспечивают Web-интерфейс, который помогает пользователям разрабатывать заказные приложения без необходимости знать инфраструктуру портала. GridPort разработан для выполнения портальных сервисов и клиентских приложений на отдельных Web-серверах. Модули инструментария GridPortal использовались для разработки научных порталов в таких областях приложений, как моделирование фармакинетики, молекулярного моделирования, кардиофизиологии и томографии.

Инструментарий сотрудничества порталов Grid [26] — это пример развитой формы сотрудничества между несколькими научными агентствами США, целью которой является поддержка общего набора компонентов и утилит для упрощения разработки порталов и интероперабельности различных порталов при использовании той же самой инфраструктуры ядра (а именно Grid Security Infrastructure (GSI) и Globus). Возможности портала следующие:

- передача файлов, загрузка в удаленный (file upload) и в локальный компьютеры (file download), передача файлов третьих лиц (перемещение файлов между различными системами хранения данных);
- запрос баз данных о спецификации ресурса/задания;
- поддержка параметров пользователя, которые содержат информацию относительно прошлых заданий, использованных ресурсов, результатов выполнения и предпочтений пользователя.

В целом, портальные архитектуры основаны на трехъярусной модели, где клиентский браузер безопасно сообщается с Web-сервером по безопасным протоколам и подключениям. Web-сервер имеет доступ к различным сервисам Grid, использующим соответствующую инфраструктуру (например, Globus). Инструментарий Globus Toolkit обеспечивает механизмы для надежной передачи заданий цензору Globus (gatekeeper), запрашивая информацию об аппаратных и программных средствах через LDAP и безопасной инфраструктуре через GSI.

3.5. Интегрированные системы. По мере появления компонентов второго поколения Grid ряд международных групп начали проекты по их объединению в когерентные системы. Эти проекты были направлены на конкретные высокопроизводительные приложения из широкого круга применений.

Одним из первых примеров такого рода систем был Cactus [27] — среда решения задач с открытым кодом. Cactus имеет модульную структуру, которая допускает выполнение параллельных приложений на широком диапазоне архитектур и возможность совместной разработки кода распределенными группами разработчиков. Cactus разработан в академической исследовательской среде, где физики и компьютерщики вместе работали над моделированием черных дыр. Cactus является входной системой ко многим внутренним сервисам параллельной файловой системы ввода/вывода HDF5 (универсальной библиотеки для хранения и форматирования файлов научных данных), научной библиотеки PETSc (набора программ и структур данных для параллельного решения

научных задач, моделируемых дифференциальными уравнениями в частных производных с использованием MPI) и продвинутых инструментальных средств визуализации. Портал содержит опции для компиляции и развертывания приложений на распределенных ресурсах.

Аналогичный европейский проект DataGrid [28], выполняемый в CERN, финансируется Европейским Союзом с целью создания крупномасштабной вычислительной и информационной сети ресурсов Grid для анализа данных научно-исследовательских работ. Первичным пилотным проектом для DataGrid явился Большой коллайдер адронов (Large Hadron Collider — LHC), который должен работать в CERN в 2005—2015 гг. и осуществить прорыв в изучении энергии, напряженности и частоты столкновений в пучках частиц. Этот рывок стал необходимым для поставки некоторых экспериментов с ранее неизвестными частицами, такими, как бозоны и сверхсимметричные кварки и лептоны. Главной проблемой, стоящей перед проектом, являются средства коллективного использования огромного количества данных, распределенных по инфраструктуре компьютерных сетей. DataGrid основывается на находящихся на стадии становления технологях Grid, которые, как ожидается, позволят развертывать крупномасштабные вычислительные среды, состоящие из распределенных коллекций файлов, баз данных, компьютеров, научных приборов и устройств.

Целями проекта DataGrid являются:

- реализация промежуточного ПО для администрирования структуры Grid, включая оценку, тестирование и интеграцию существующих средств промежуточного ПО и разработку новых средств программного обеспечения;
- развертывание крупномасштабной системы испытаний;
- обеспечение демонстрационных приложений промышленного качества.

Работа сосредоточена на распределенной обработке крупномасштабных приложений в области физики высоких энергий, биоинформатики, наук о земле. DataGrid разрабатывается на основе Globus и включает следующие компоненты:

- Язык описаний заданий (JDL) — сценариев для описания параметров заданий.
- Интерфейс пользователя (UI) — посылает работу брокеру ресурсов RB и принимает результаты.
- Брокер ресурсов (RB) — находит и выбирает целевой вычислительный элемент (Computing Element — CE).
- Сервис представления заданий (JSS) — представляет задание целевому CE.
- Регистрация и бухгалтерский учет (L&B) — запись информации о статусе заданий.
- Информационная служба Grid (GIS) — индекс информации о состоянии структуры Grid.
- Каталог репликаций — список наборов данных и их дубликатов, находящихся на запоминающих элементах (Storage Elements — SE).

Еще один европейский проект UNICORE (UNIform Interface to COmputer REsources) [29] — это проект, финансируемый Министерством образования и науки Германии. Цели разработки UNICORE включают единый и легкий в использовании графический интерфейс пользователя, открытую архитектуру, основанную на понятии абстрактного задания, целостную (непротиворечивую) архитектуру защиты, минимальную зависимость от локальных административных процедур, эксплуатацию существующих и находящихся на стадии становления технологий, основанных на стандарте Java и Web-технологиях. UNICORE обеспечивает интерфейс для подготовки заданий и безопасного их представления на распределенные ресурсы суперЭВМ.

Распределенные приложения в UNICORE определены как многосторонние, где различные части могут вы-

полняться разными компьютерными системами асинхронно или синхронно-последовательно. Задание в UNICORE содержит многостороннее приложение, оснащенное информацией о системе, для которой оно предназначено к выполнению, о необходимых требованиях ресурса и зависимостях между различными частями. Структурно задание UNICORE — это рекурсивный объект, содержащий группы заданий и задач. Задания UNICORE и группы заданий несут информацию о системе назначения для содержащихся задач. Задание есть модуль, который сводится к пакетному заданию для системы назначения.

Главные компоненты UNICORE: агент подготовки заданий (JPA); контроллер монитора заданий (JMC); сервер https UNICORE, также называемый шлюзом (Gateway); супервизор сетевых заданий (NJS); графический интерфейс пользователя, основанный на Java-аплетах, с интерактивной справкой и средством помощи.

Клиент UNICORE дает возможность потребителю создавать, представлять и управлять заданиями с любой рабочей станции или персонального компьютера, подключенных к Internet. Клиент соединяется с UNICORE через шлюз, который подтверждает подлинность клиента и пользователя перед контактированием с серверами UNICORE, которые в свою очередь управляют представленными UNICORE заданиями. Задания, предназначенные для локальных компьютеров, выполняются на их пакетных подсистемах, те же, которые будут выполнены на удаленных узлах, передаются на просмотр шлюзу. Все необходимые передачи данных и синхронизации выполняются серверами. Серверы также сохраняют информацию о статусе и результатах работы, передавая их клиентам по запросу пользователя. Компоненты третьей стороны, такие как Globus, могут быть интегрированы в структуру UNICORE для расширения его функциональности.

UNICORE широко используется и разрабатывается для проекта EuroGrid [30], финансируемого Европейской Комиссией. Этот проект направлен на изучение возможностей и специфических требований использования Grid в таких научных и промышленных направлениях, как биология, метеорология и автоматизация проектирования. Цели проекта EuroGrid включают поддержку инфраструктуры программного обеспечения EuroGrid, разработку программных компонентов и прототипов программ распределенного моделирования в различных прикладных областях (биомолекулярное моделирование, прогноз погоды, интегрированное моделирование в автоматизации конструирования, структурный анализ и обработка данных в реальном масштабе времени).

WebFlow [31] — еще одно вычислительное расширение модели Web, которое может быть использовано как основа для распределенной широкомасштабной обработки. Главная цель построения WebFlow состояла в том, чтобы сформировать единый «бесшовный» каркас для опубликования и многократного использования вычислительных модулей на Web так, чтобы конечные пользователи с помощью браузера сети могли участвовать в создании распределенных приложений, использующих модули WebFlow как визуальные компоненты, а редакторы как визуальные средства установления подлинности. WebFlow имеет трехъярусную основанную на Java архитектуру, которая может рассматриваться как визуальная система потока данных. Входные аплеты используются для авторизации, визуализации и управления средой. WebFlow использует основанный на сервлетах промежуточное ПО для управления и взаимодействия с прикладными модулями, такими, как унаследованные коды для баз данных или высокопроизводительного моделирования.

WebFlow аналогичен Web; Web-страницы можно сравнить с WebFlow-модулями, а гиперсвязи, которые свя-

зывают Web-страницы, — с межмодульными каналами потоков данных. Разработчики содержимого WebFlow создают и публикуют модули, подсоединяя их к Web-серверам. Интеграторы приложений используют визуальные инструментальные средства, чтобы связать выходы исходных модулей со входами модулей назначения, формируя таким образом распределенные вычислительные графы (или вычислительные Web) и публикуют их как составные объекты модулей WebFlow. Пользователь инициирует эти вычислительные Web, переходя по соответствующим гиперссылкам, настраивая вычисление в терминах доступных параметров или используя высокоуровневые инструментальные средства для визуальной авторизации графов. Прикладные модули WebFlow реализованы с используованием Globus Toolkit, в частности MDS, GRAM и GASS.

3.6. Одноранговые вычисления. Одним из подходов к решению проблем масштабирования является децентрализация. Традиционная клиент-серверная модель может стать узким местом в повышении производительности и единственным источником отказов, хотя все еще остается превалирующей, потому что децентрализация несет с собой собственные проблемы. Тем не менее одноранговые (Peer-to-Peer — P2P) вычисления [32] (в Napster [33] и Gnutella [34]) и Internet-вычисления (в SETI@home [8] и Entropia [35]) являются примерами весьма общих вычислительных структур, которые пользуются преимуществом глобально распределенных ресурсов.

В вычислениях P2P компьютеры разделяют данные и ресурсы, такие как процессорное время и емкость долговременной памяти, через Internet или частные сети. Компьютеры могут также сообщаться непосредственно и управлять вычислительными заданиями без использования центральных серверов. Это позволяет масштабировать вычисления P2P более эффективно, чем традиционные клиент-серверные системы, которые должны развернуть для этого серверную инфраструктуру для обеспечения возможностей расширения. Децентрализация клиента и сервера одновременно является привлекательной не только относительно масштабируемости, но и отказоустойчивости. Однако имеются некоторые препятствия на пути широкого принятия систем P2P, которые сводятся к следующему.

1. От персональных компьютеров и рабочих станций, используемых в сложных приложениях P2P, потребуются большие вычислительные возможности для выполнения дополнительной нагрузки по связи и защите информации, которые обычно несут серверы.

2. Защита является серьезной проблемой, поскольку компьютеры для приложений P2P требуют доступа к ресурсам других компьютеров (памяти, жестким дискам и т.д.). Загрузка файлов из других компьютеров делает системы уязвимыми со стороны вирусов.

3. Системы P2P должны работать с разнородными ресурсами, использующими различные компоненты работы с сетями и операционные системы.

4. Одним из самых больших проблем P2P-вычислений является возможность устройств находить друг друга в вычислительной структуре, в которой отсутствует центральное управление.

Идеи P2P начинают привлекать и больших производителей. В 2001-м г. Sun Microsystems возвестила о начале проекта JXTA [36] с открытым кодом для инфраструктуры и приложений P2P.

Таким образом, второе поколение базового программного обеспечения для Grid в своем развитии перешло от ранних систем типа Globus-GT1 и Legion, специализированных для конкретных потребностей больших и высокопроизводительных приложений, к более универсальным и открытым средам, таким, как Globus-GT2. Вместе с базовым программным обеспечением во втором поколении также разработан ряд сопровождающих инструментальных средств и утилит для сервисов

верхнего уровня, которые охватили планировщиков ресурсов и брокеров, а также предметно-ориентированные интерфейсы пользователей и порталы. В течение этого периода появилась и техника одноранговых систем.

4. Эволюция Grid: третье поколение

Второе поколение обеспечило интероперабельность, которую было существенно достичь для крупномасштабных вычислений. По мере дальнейших исследований возникли и другие подходы к разработкам проблем построения Grid. Стало очевидным, что для создания новых приложений Grid желательно иметь возможность многократно использовать существующие компоненты и информационные ресурсы и собирать эти компоненты некоторым гибким способом. Таким образом, усилилось внимание к *сервисно-ориентированной модели и использованию метаданных,* которые стали двумя ключевыми характеристиками систем третьего поколения. Они тесно связаны между собой, ибо сервисно-ориентированный подход имеет непосредственное значение для структуры информации. Гибкая сборка ресурсов Grid в приложения требует информации о функциональных возможностях, доступности и интерфейсах различных компонентов, и эта информация должна иметь согласованное истолкование, которое может быть обработано машиной.

Если традиционно Grid описывался в терминах крупномасштабных данных и вычислений, то в фокусе третьего поколения оказались новые и более общие понятия. В частности, если термины “распределенное взаимодействие” и “виртуальная организация” были приняты в основополагающей работе по исследованию Grid второго поколения [37], то для третьего поколения стало характерным более целостное представление вычислений Grid. Оно затрагивает не только технологию вычислений, но и всю инфраструктуру и приводит к новому понятию E-науки (новые способы ведения исследований и организации труда ученого). Как заметил J. Fox [38], ожидаемое использование вычислительных средств с массовым параллелизмом — только часть появляющейся панорамы Grid, у которой имеется намного больше пользователей.

Автоматизации вычислительных процессов в системах третьего поколения придается гораздо более общий и глубокий смысл. В частности, пользователи непосредственно больше не должны иметь дело с немасштабируемостью и неоднородностью, отслеживание этих свойств делегируются процессам (в частности, посредством сценариев-скриптов), что ведет к автономности этих процессов в пределах всей системы. Это подразумевает потребность в координации, которая, в свою очередь, должна быть определена программно на различных уровнях описания процессов. Точно так же большая вероятность неисправности подразумевает потребность в автоматическом восстановлении, поскольку конфигурирование и наладка уже не могут оставаться ручными операциями. Эти требования становятся похожими на самоорганизацию в биологических системах, состоящих из автономных подсистем. Выдвинутая IBM концепция системы автономных вычислений [39] содержит следующие свойства:

1) нуждается в детальном познании своих компонентов и их состояния;

2) должна конфигурировать и реконфигурировать себя динамически;

3) стремится оптимизировать свое поведение, чтобы достичь своей цели;

4) способна восстанавливаться после неисправности;

5) защищает себя от атак;

6) должна знать свою среду;

7) реализует открытые стандарты;

8) оптимизирует использование ресурсов.

Разрабатываемые ныне системы Grid третьего поколения уже начинают выявлять многие из этих свойств.

4.1. Сервисно-ориентированная архитектура. К 2001 году свойства архитектуры Grid уже были присущи многим проектам; например, в проекте Information Power Grid [40] показан обширный набор сервисов, размещаемых по слоям. В это же время приобрела популярность модель Web-сервисов с обещанием стандартов в поддержку сервисно-ориентированного подхода. К тому времени исследовательское сообщество агентно-ориентированных вычислений уже выполнило обширную работу в этой области: современные программные агенты уже стали как поставщики, потребители и брокеры сервисов. В целом было очевидно, что сервисно-ориентированная парадигма обеспечивает гибкость, требуемую для третьего поколения Grid. И, таким образом, оказалось, что три технологии (агентные, Grid-вычислений и Web-сервисов) тесно связаны друг с другом и вместе составляют три источника и три составляющих технологии E-науки.

4.1.1. Web-сервисы. Создание стандартов Web-сервисов — это основа деятельности консорциума Всемирной паутины W3C, состоящая из нескольких находящихся в стадии становления стандартов, включающих:

- SOAP (протокол XML) обеспечивает конверт (оболочку), который инкапсулирует данные XML для передачи через инфраструктуру Web (например, по протоколу HTTP с помощью кэшей и промежуточных процессов) в соответствии с правилами для удаленного вызова и механизмами сериализации, основанными на типах данных XML Schema. SOAP развивается W3C в сотрудничестве с рабочей группой Internet Engineering (IETF).
- Язык описаний Web-сервисов (WSDL) [41] описывает сервисы в XML, используя XML Schema; имеется также отображение на язык описаний ресурсов Resource Description Framework (RDF). В некотором роде WSDL подобен языку описания интерфейсов IDL.
- Универсальное описание обнаружения и интеграции (UDDI) [42] представляет собой спецификацию для распределенных реестров Web-сервисов наподобие сервисов "Желтых страниц" — адресных справочников. UDDI поддерживает парадигму "publish, find and bind" (издать, найти и связать), где поставщик услуг описывает и оповещает о подробностях обслуживания в каталоге; потребитель услуги делает запрос к системному реестру, чтобы найти провайдера услуги; эти услуги "связываются" с использованием технических деталей от UDDI.

Следующие стандарты Web-сервисов, привлекающие интерес, находятся на уровне процессов. Например, язык потоков данных Web-сервисов (Web Services Flow Language — WSFL) [43] является предложением от IBM, которое определяет последовательности работ как комбинации Web-сервисов и дает возможность последовательностям работ самим выступать как сервисы. Предложение XLANG [44] от Microsoft поддерживает комплексные транзакции, которые могут включать множество Web-сервисов. В дополнение к программному обеспечению для самих Web-сервисов сделаны важные шаги в конструировании систем Web-сервисов. Например, каркас моделирования Web-сервисов (WSMF) обеспечивает концептуальную модель для разработки и описания Web-сервисов, основанных на принципах максимального расщепления и масштабирования медиаторных сервисов [45].

Web-сервисы тесно вплетены в структуру Grid третьего поколения: они поддерживают сервисно-ориентированный подход и следуют стандартам для обеспечения описаний информации, в том числе самих сервисов. Фактически WSDL описывает, как взаимодействовать с сервисом, а не его функциональные возможности. Дальнейшие усилия по описанию сервисов затрагивают агентные технологии, например DAML-S [46].

4.1.2. Каркас открытой архитектуры сервисов Grid (OGSA). Каркас открытой архитектуры сервисов Grid (OGSA) — это совместный взгляд

Globus и IBM на процесс слияния Web-сервисов и вычислений Grid [13]. OGSA направлена на создание, ведение и применение наборов сервисов, поддерживаемых виртуальными организациями. Сервис определяется как объект в сети, который обеспечивает возможности вычислительных ресурсов, ресурсов памяти хранения, сетей, программ и баз данных. Это позволяет подходу Web-сервисов удовлетворять некоторым требованиям Grid. Таковыми являются, например, стандартные интерфейсы, определенные в OGSA:

- *обнаружение*: клиентам требуются механизмы для обнаружения доступных сервисов и определения их характеристик, чтобы они могли конфигурировать себя и запросы к этим сервисам соответственно;
- *динамическое создание сервисов*: стандартный интерфейс Factory и семантика, которую любой сервис создания сервиса должен обеспечивать;
- *администрирование жизненного цикла*: в системе, в которой сочетаются сервисы с состояниями и без таковых, должны быть предусмотрены механизмы восстановления сервисов и состояний после неправильных действий;
- *уведомления*: динамические распределенные сервисы должны уметь асинхронно уведомлять друг друга относительно тех изменений, которые происходят с их состояниями;
- *управляемость*: должны обеспечиваться действия по администрированию и контролю большого количества сервисов Grid;
- *простая среда пребывания* (hosting) — набор ресурсов, находящийся в одном административном домене и поддерживающий первичные средства для обслуживания пользователя, например сервер приложений J2EE, система Microsoft.NET или кластер Linux.

Больше всего архитектура OGSA повлияла на такие части Globus, как протокол распределения и управления ресурсами Grid (GRAM), сервис метакаталогов (MDS-2), используемый для обнаружения информации, регистрации, моделирования данных и локального системного реестра, и инфраструктуру безопасности Grid (GSI), которая поддерживает условия одиночного входа, ограниченного делегирования и отображения мандатов. Ожидается, что будущие реализации инструментария Globus будут основаны на архитектуре OGSA.

4.1.3. Агенты. Web-сервисы обеспечивают средства интероперабельности, являющиеся ключевыми для вычислений на Grid, и поэтому OGSA является существенным элементом стратегии, которое адаптирует Web-сервисы к Grid и приближает потребность в приложениях Grid. Однако сами Web-сервисы не дают новых решений для главных проблем крупномасштабных распределенных систем, они даже не обеспечивают новых методов их разработки. Поэтому следует рассмотреть другие сервисно-ориентированные модели, основанные на агентах вычисления [47], которые являются существенным дополнением сервисно-ориентированной модели Grid.

Парадигма агентных вычислений представляет перспективу программных систем в использовании объектов, которые обычно имеют следующие свойства, обозначаемые как слабые агенты [48]:

1) *автономность* — агенты функционируют без вмешательства извне и имеют некоторый контроль над своими действиями и внутренним состоянием;

2) *социальная способность* — агенты взаимодействуют с другими агентами, используя язык коммуникаций агентов;

3) *реактивность* — агенты воспринимают и реагируют на их среду;

4) *про-активность (инициативность)* — агенты выявляют управляемое целью поведение.

Агентные вычисления особенно подходят для динамически изменяющейся среды, где их автономность дает возможность адаптироваться к изменяющимся обстоятельствам. Это харак-

терное свойство для третьего поколения Grid. Один из методов для достижения этого свойства — переговоры между компонентами, и имеется существенный задел в исследованиях техники переговорных механизмов [49]. В частности, методы, основанные на рынкоподобных механизмах, имеют весомое значение для вычислительной экономики, которая возникает в приложениях Grid.

Следовательно, можно рассматривать Grid как ряд взаимодействующих компонентов и информацию, которая передается в этих взаимодействиях. Последнюю можно отнести к нескольким категориям. Одна из них — информация контекста предметной области. Другие типы включают следующую информацию: о компонентах и их функциональных возможностях в пределах предметной области; о связях с компонентами; об общем потоке работ и конкретных потоках как частей общего.

Компоненты должны быть связаны стандартным способом для обеспечения интероперабельности между ними на основе согласованных общих словарей. В языках взаимодействия агентов (ACL) эти вопросы рассматриваются на формальной основе. В частности, Организация по вопросам интеллектуальных физических агентов (Foundation for Intelligent Physical Agents — FIPA) разрабатывает подходы к определению семантики для этой информации на основе интероперабельности, а также стандарты на программное обеспечение для разнородных и взаимодействующих агентов и агентно-ориентированных систем, включая обширные спецификации. В абстрактной архитектуре FIPA агенты взаимодействуют, обмениваясь сообщениями, которые представляют собой речевые акты, закодированные в языке их взаимодействия; сервисы предоставляют услуги агентам (включая службы каталогов и передачи сообщений) и могут быть реализованы или как агенты, или как вызываемое программное обеспечение, к которому обращаются с использованием программного интерфейса (например, в Java, C++ или IDL).

Таким образом, можно идентифицировать обмен информацией между агентами и обращения к каталогам как форматы информации, распознаваемые на инфраструктурном уровне.

4.2. Соотношение Web и Grid. Важно понимать причины быстрого внедрения Web и то, как это может воздействовать на разработку Grid, которая имеет такие же устремления в смысле масштаба и развертывания. Первая причина — это простота. HTTP и HTML внесли не так много нового для современного пользователя, и это облегчило их массовое распространение. Следует однако понимать большую разницу между Web и Grid: несмотря на крупный масштаб Интернета количество хост-машин, вовлеченных в типичную транзакцию на Web, все еще незначительно и намного меньше, чем предусмотрено для многих приложений Grid.

4.2.1. Web как информационная инфраструктура Grid. Первоначально Web была создана для распределения информации в контексте E-науки в CERN. Естественно выяснить, удовлетворяет ли сейчас эта архитектура распределения информации требованиям Grid. При этом возникают следующие вопросы:

- *Управление версиями.* Популярная парадигма публикаций на Web предполагает непрерывное обновление страниц без контроля версий, и сама инфраструктура Web явно не поддерживает такую возможность.

- *Качество обслуживания.* Гиперссылки являются встроенными и постоянными, они ненадежны и бесполезны, если изменяется сервер, местоположение, название или содержание документа предназначения. Ожидания непротиворечивости ссылок низки, и E-наука может требовать более высокого качества обслуживания.

- *Происхождение информации.* В Web нет никакого стандартного механизма для обеспечения юридически

значимого свидетельства того, что документ был представлен на Web в означенное время.

• *Цифровое управление правами*. Е-наука требует специфических функциональных возможностей относительно управления цифровым содержанием, включая, например, защиту от копирования и управление интеллектуальной собственностью.

• *Надзор*. Многое из инфраструктуры Web сосредоточено на технике доставки информации, а не на средствах создания и управления содержанием. Проектировщики инфраструктуры Grid должны обратить внимание на поддержку метаданных с самого начала.

Рассмотрим, как некоторые из этих вопросов решаются в других областях. Например, в индустрии мультимедийной информации также требуется поддержка для цифрового управления правами. Ее элементы включают декларацию, идентификацию, обработку содержания, управление интеллектуальной собственностью и защиту. Авторское право (Authoring) — еще один важный момент, особенно совместное авторское право. Действие на Web-документ Distributed Authoring and Versioning (WebDAV [50]) предписывает определить расширения протокола HTTP, необходимые для интероперабельного использования распределенных инструментов авторского права на Web при поддержке потребностей пользователя. В результате, хотя Web и предоставляет эффективную среду для транспортировки информации, это не обеспечивает всесторонней информационной инфраструктуры для Е-науки.

4.2.2. Выражение содержания и метасодержания. Web все больше становится инфраструктурой для распределенных приложений, где скорее происходит обмен информацией между программами, нежели чтение ее человеком. Такой информационный обмен обеспечивается семейством рекомендаций XML от W3C. XML предназначен для разметки документов и не имеет никакого установленного словаря тегов; они определены для каждого приложения и используют Document Type Definition (DTD) или XML Schema. RDF — это стандартный способ выражения метаданных, особенно ресурсов на Web, хотя им можно воспользоваться для представления структурированных данных вообще. Использование XML и RDF делает возможным стандартное выражение содержания и метасодержания. Появляются дополнительные наборы инструментов для работы с этими форматами, и это увеличивает поддержку со стороны других инструментов. Все вместе это обеспечивает инфраструктуру для информационных систем третьего поколения Grid. W3C опубликовал документ [51], в котором рассмотрена перспективная технология Semantic Web, определяемая как расширение нынешней сети Web, при которой информация имеет четко определенное значение, предоставляющее лучшие возможности для сотрудничества людей и компьютеров. Главное, что несет эта технология, — это идея наличия данных на Web, определенных и связанных таким способом, который позволяет использовать их для более эффективного обнаружения, автоматизации, интеграции и повторного использования в различных приложениях.

Таким образом, Web может достигнуть раскрытия своего полного потенциала, если станет местом совместного использования и обработки автоматизированными инструментами и людьми, а Semantic Web предназначена сделать для представления знаний то, что Web сделала для гипертекста. Исследовательская программа языка разметки агентов DAML (DARPA Agent Markup Language) [52] привносит в Semantic Web коммуникации агентов. DAML расширяет XML и RDF с помощью онтологий — мощного способа описания объектов и их отношений.

4.3. Сравнение Web и Grid. Состояние развития Grid сегодня напоминает Web десятилетие тому назад с ограниченным распространением, в значительной степени управляемым

энтузиастами от науки, с появляющимися стандартами и некоторыми попытками коммерческой деятельности. То же самое можно было бы сказать и относительно Semantic Web. Тем временем коммуникации в Web измененились от типа "машина — человек" (с помощью HTML) к типу "машина — машина" (с помощью XML), и это оказалось именно той инфраструктурой, которая необходима для Grid. Связанная с этим парадигма Web-сервисов, кажется, обеспечивает соответствующую инфраструктуру для Grid, хотя требования Grid уже расширяют эту модель.

Из этих сравнений напрашивается вывод, что развертывание Grid будет следовать той же экспоненциальной модели, что и рост Web. Однако типичное приложение Grid может вовлекать огромное количество процессов, взаимодействующих скоординированным способом, в то время как типичная транзакция на Web сегодня все еще использует лишь небольшое количество ресурсов (например, сервер, кэш, браузер). Достижение желательного поведения от крупномасштабной распределенной системы ставит важные технические проблемы, которые сама Web не должна рассматривать, хотя Web-сервисы заставляют нас обратиться к ним.

Web обеспечивает инфраструктуру для Grid. Обратно, можем спросить, что Grid предлагает для Web. Как Web-приложение Grid поднимает некоторые вопросы, мотивирующие развитие технологий Web, например усиление Web-сервисов в OGSA, которые могут хорошо выходить за пределы приложений. Grid также обеспечивает высокопроизводительную инфраструктуру для различных аспектов Web-приложений, например в поиске, выявлении зависимостей данных, переводе и представлении мультимедийной информации.

4.4. Semantic Grid. Понятия Grid и Semantic Grid имеют много общего, но могут различаться в акцентах: Grid традиционно сосредоточена на вычислениях, в то время как Semantic Grid — больше на выводе, доказательстве и проверке. Grid, которая теперь строится в своем третьем поколении, ведет к тому, что называем Semantic Grid [53], а именно это Grid, основанная на использовании метаданных и онтологий. Третье поколение Grid идет по пути, где *информация* представляется, запоминается, становится доступной и совместно используемой и поддерживается. Такая информация понимается как данные, имеющие значение. Предполагается, что следующее поколение будет иметь дело уже со *знаниями*, которые приобретаются, используются, представляются, публикуются и поддерживаются, чтобы помочь Е-ученым достигать их специфических целей. Знание понимается как информация, примененная для достижения цели, решения проблемы или принятия решения. Semantic Grid вовлекает все три концептуальных слоя Grid: знание, информация и вычисление/данные. Эти дополнительные слои в конечном счете обеспечат богатый, бесшовный и распространяющийся доступ к глобально распределенным гетерогенным ресурсам.

4.5. Новые формы для научных исследований — живые информационные системы. Третье поколение Grid акцентируется на распределенном сотрудничестве. Один из аспектов совместной работы пользуется идеей «колаборатории», или центра без стен, в котором национальные исследователи могут выполнять исследования безотносительно к географическому местоположению путем взаимодействия с коллегами, совместно используя инструментарий, данные и вычислительный ресурс и обращаясь за информацией к цифровым библиотекам. Такое представление фактически превращает информационные приборы, какими являются компьютеры и сетевая инфраструктура, в лабораторные установки, которые могут, например, включать электронные журналы и другие портативные устройства.

В Web сейчас широко распространена подача информации, что является

мощным стимулом для создании кругов общения. Однако такая парадигма взаимодействия по существу реализует принцип "издания один у одного", подкрепленный службами электронной почты и групп новостей, которые также поддерживают *асинхронное сотрудничество*. Несмотря на это основная инфраструктура Интернета, однако, полностью способна к поддержке *живых* (в реальном масштабе времени) информационных услуг и *синхронного сотрудничества:* живые данные от экспериментального оборудования; живое видео с помощью Web-камер через одностороннюю или широковещательную передачи; видеоконференции; чаты; системы моментального информирования; многопользовательские диалоги и игры; колаборативные виртуальные среды. Все они играют определенную роль в поддержке Е-науки, *непосредственно связывая людей* вне сцен и процессов инфраструктуры. В частности, они поддерживают расширение Е-науки по направлению к новым сообществам, преодолевая установленные организационные и географические границы. Акцент делается на обеспечении распределенного сотрудничества, охватывающего все более и более интеллектуальные (smart) сферы работы Е-ученых [54]. Эта область исследований подпадает под направление "Перспективные колаборативные среды" Рабочей группы Глобального форума Grid (ACE Grid), в котором рассматриваются среды коллективной работы и вездесущие (ubiquitous) вычисления.

Другим примером живых информационных систем являются сети доступа [55] — собрания ресурсов, поддерживающие сотрудничество людей в Grid, в частности крупномасштабные распределенные встречи и обучение. Ресурсы включают показ мультимедийной информации и взаимодействие, особенно через комнаты видеоконференций "группа-на-группу", а также интерфейсы к промежуточному ПО и средам визуализации. Узлы Grid для доступа обладают специальными средствами обслуживания, которые поддерживают высокое качество звуковых и видеотехнологий, необходимых для эффективной работы пользователей. Во время встречи происходит живой обмен информацией, что выдвигает на первый план информационные аспекты. Это приводит к изменению метаданных, генерируемых автоматически программным обеспечением и устройствами, что может быть использовано для обогащения конференции и записи для более позднего воспроизведения. Могут понадобиться новые формы обмена информацией, чтобы обеспечить крупный масштаб встреч, таких как распределенный опрос и голосование.

Еще один источник живой информации — замечания, принятые членами встречи, или аннотации, которыми они сопроводили существующие документы. Как и выше, их можно совместно использовать и записывать для обогащения содержания встреч. Особенность текущих технологий сотрудничества заключается в том, что могут легко быть созданы и без помех частные обсуждения, которые также обеспечивают обогащенное содержание. В видеоконференциях живое видео и звук обеспечивают *эффект присутствия* для удаленных участников (для них также можно установить другие формы присутствия, например использование искусственных образов (олицетворений — avatars) в колаборативной виртуальной среде).

Выводы

Эволюция Grid является непрерывным процессом, строго не определенным, который лучше характеризовать как философию, а не технологию. В статье проанализированы первые три поколения Grid:

- системы первого поколения развивали каждая собственные решения для совместного использования высокопроизводительных вычислительных ресурсов;
- системы второго поколения ввели промежуточное ПО, чтобы решить

проблемы масштабирования и разнородности, с акцентом на крупномасштабные задачи, использующие большую вычислительную мощность и большие объемы данных;

• системы третьего поколения принимают сервисно-ориентированный подход и более целостное представление об инфраструктуре Е-науки; они используют метаданные и могут выявлять автономное поведение.

Общее представление о современной Grid-системе состоит в трехслойной структуре, составленной из вычислений/данных, информации и знаний. Даже при том, что слой вычислений/данных является, возможно, наиболее зрелым в смысле времени, опыта и использования программного обеспечения, он все еще испытывает недостаток во многих существенных аспектах, необходимых для беспрепятственного, распространяющегося и безопасного использования ресурсов системы. Некоторое количество этих аспектов теперь рассматривается в слоях информации и знаний эволюционирующих Grid-систем.

Следующие вопросы представляются перспективными для дальнейших исследований [12].

• Информационные сервисы — механизмы, которые используются для накопления и поддержки информации о ресурсах Grid, требуемых для обеспечения расширяемых, быстрых, надежных, безопасных и масштабируемых услуг.

Информация о ресурсах — всесторонняя информация о Grid необходима для правильной ее работы. Она охватывает диапазон от данных безопасности до требований к приложениям и от именования ресурсов до профилей пользователей. Важно, чтобы вся эта информация могла быть понята, интерпретирована и использована всеми сервисами, которые ее потребуют.

• Обнаружение ресурсов — при заданном уникальном имени ресурса или его характеристики должен существовать механизм его нахождения в глобально распределенной системе. Сервисы являются ресурсами: одни могут храниться постоянно, другие — быть преходящими, а некоторые — создаваться по требованию.

• Синхронизация и координация — как организовать сложную последовательность вычислений при разнообразии ресурсов, учитывая свойства слабо- и тесносвязанных распределенных систем. Это может потребовать описаний процессов и инфраструктуры, основанной на событиях, что повлечет за собой необходимость планирования на различных уровнях, включая метапланирование и потоки работ.

• Отказоустойчивость и надежность — операционные среды должны справляться с отказами программного обеспечения и компонентов аппаратных средств ЭВМ, а также с проблемой доступа, приспосабливая для этого систему обработки исключений, которая является необходимой в таких динамических многопользовательских многоорганизационных системах.

• Безопасность — подлинность, авторизация, гарантия и бухгалтерский учет должны быть установлены и функционировать в контексте увеличивающегося масштаба и автоматизации. Например, пользователь делегирует привилегии процессам, которые, в свою очередь, распространяют некоторые привилегии далее.

• Параллелизм и непротиворечивость — потребность обслуживать соответствующий уровень непротиворечивости данных в параллельной гетерогенной среде. Более слабая непротиворечивость может быть достаточной для некоторых приложений.

• Производительность — потребность справляться с нелокальным доступом к ресурсам через кэширование и дублирование. Перемещение кода или сервиса к данным (возможно, со скриптами или мобильными агентами) является привлекательным и может породить новые проблемы.

• Разнородность — потребность работать со множеством аппаратных средств ЭВМ, программного обеспече-

ния и информационных ресурсов в разных организациях с различными административными структурами.

• Масштабируемость — системы должны быть способными повысить количество и объем услуг и приложений без потребности в ручном вмешательстве. Это требует автоматизации, идеально, самоорганизации.

В информационном слое, где некоторые из технологий доступны уже сегодня (хотя и в ограниченной форме), многие вопросы все еще требует исследования. Они включают:

• Проблемы, касающиеся типов содержимого E-науки, кэширования для новопорожденного содержания. Как инфраструктура Web ответит на различные шаблоны доступа, вытекающие из автоматизированного доступа к информационным источникам? Проблемы сбережения и приумножения содержания E-науки.

• Цифровое управление правами в контексте E-науки (по сравнению, например, с мультимедийной информацией и E-торговлей).

• Происхождение. Источник должен запоминаться, чтобы облегчить повторное использование информации, повторение экспериментов или как свидетельство того, что данная информация существовала в некоторое время.

• Создание и управление метаданными, обеспечение инструментов для поддержки метаданных.

• Описания сервисов и инструменты для работы с ними. Как лучше всего описывать сервисно-ориентированную архитектуру?

• Описание и воплощение потоков работ. Инструменты для работы с описаниями.

• Адаптация и персонализация. Сколько знания может быть приобретено с помощью метаданных и как они могут использоваться?

• Колаборативные инфраструктуры для более широких сообществ, включая взаимодействие между учеными, содержанием E-науки и визуализацией, и связывание интеллектуальных лабораторий.

• Использование метаданных в колаборативных событиях, особенно живых метаданных; установление схем метаданных для поддержки сотрудничества на встречах и в лабораториях.

• Представление информации, использующей новые формы устройств, например, для ученых, работающих в полевых условиях.

• Представление информации об основной структуре Grid, как того требует приложение, например, для планирования ресурса и контроля.

Semantic Web станет областью, где данные снабжены богатым контекстом и превращаются в информацию. Эта информация будет тогда общей (разделяемой) и ее можно будет обрабатывать виртуальным организациям, чтобы достичь определенных целей. Такая активная (отзывчивая — actionable) информация составляет знание. Следовательно, слой знания является ключевым на пути к следующей стадии в развитии от Grid к Semantic Grid.

1. *Catlett C., Smarr L.* Metacomputing // Comm. ACM. — 1992. — № 6. — P. 44–52.
2. *The Grid*: Blueprint for a New Computing Infrastructure / Eds. I. Foster, C. Kesselman. — San Francisco: Morgan Kaufmann Publishers. — 1999. — 260 p.
3. *Foster I., Kesselman C., Tuecke S.* The Anatomy of the Grid: Enabling Scalable Virtual Organizations // Intern. J. of High Performance Computing Applications. — 2001. — **15**(3). — http://www.globus.org/research/papers/anatomy.pdf
4. *Grid* Computing: Making the Global Infrastructure a Reality / Eds. F. Berman, G. Fox, T. Hey. — Chichester: John Wiley & Sons. — 2003. — 1060 p.
5. *National* Collaboratories: Applying Information Technology for Scientific Research. — Washington, D.C.: National Academy Press. — 1993. — http://www.nap.edu/books/0309048486/html
6. FAFNER. — http://www.npac.syr.edu/factoring.html
7. Software Infrastructure for the I-WAY High Performance Distributed Computing Experiment / I. Foster, J. Geisler, W. Nickless, W. Smith, S. Tuecke // Proc. 5th IEEE Symposium on High Performance Distributed Computing. — 1997. — P. 562–571.

8. *SETI@Home*. — http://setiathome.ssl.berkeley.edu/
9. *Distributed.Net*. — http://www.distributed.net/
10. *Foster I., Kesselman C.* Globus: A Metacomputing Infrastructure Toolkit // Intern. J. of Supercomputer Applications. — 1997. — **11**(2). — P. 115–128.
11. *Grimshaw A., Wulf W.* The Legion Vision of a Worldwide Virtual Computer // Comm. ACM. — 1997. — **40**(1). — P. 39–45.
12. *The Evolution* of the Grid / D. De Roure, M.A. Baker, N.R. Jennings, N.R. Shadbolt // [4], 2003. — P. 65 — 100.
13. *The Physiology* of the Grid: Open Grid Services Architecture for Distributed Systems Integration / I. Foster, C. Kesselman, J. Nick, S. Tuecke. — 2002. — http://www.globus.og/ /research/papers/ogsa.pdf
14. *Baker M.A., Fox G.C., Yau H.W.* A Review of Cluster Management Software, NHSE Review. — 1996. — **1**, №. 1. — http://www.nhse.org/NHSEreview/CMS/
15. *Condor.* — http://www.cs.wisc.edu/condor/
16. *Portable* Batch System. — http://www.openpbs.org/
17. *Sun* Grid Engine. — http://www.sun.com/ /software/gridware/
18. *Platform* Computing. — http://www.platform.com
19. *Rajasekar A. K., Moore R. W.* Data and Metadata Collections for Scientific Applications // European High Performance Computing Conf. — Amsterdam: Holland, 2001. — P. 56–67.
20. *Abramson D., Giddy J., Kotler L.* High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid? // Intern. Parallel and Distributed Proc. Symposium (IPDPS), IEEE Computer Society Press, 2000. — P. 289–302.
21. *Buyya R., Giddy J., Abramson D.* An Evaluation of Economy-based Resource Trading and Scheduling on Computational Power Grids for Parameter Sweep Applications // The Second Workshop on Active Middleware Services (AMS 2000), HPDC 2001, Pittsburgh: Kluwer Academic Press, USA, 2000. — P. 31–33.
22. *ActiveSheets*: Super-Computing with Spreadsheets / D. Abramson, P. Roe, L. Kotler, D. Mather // 2001 High Performance Comp. Symp. (HPC'01): Advanced Simulation Technologies Conf., 2001. — P. 29–40.
23. *Buyya R.* The Virtual Laboratory Project: Molecular Modeling for Drug Design on Grid // IEEE Distibuted Systems Online. — 2001. — **2**, No. 5. — http://www.buyya.com/vlab/
24. *HotPage*. — https://hotpage.npaci.edu/
25. *SDSC* GridPort Toolkit. — http://gridport.npaci.edu/
26. *NLANR* Grid Portal Development Kit. — http://dast.nlanr.net/Features/GridPortal/
27. *Supporting* Efficient Execution in Heterogeneous Distributed Computing Environments with Cactus and Globus / G. Allen, T. Dramlitsch, I. Foster, N. Karonis, M. Ripeanu, E. Seidel, B. Toonen. — Supercomputing 2001. — P. 12–22.
28. *The DataGrid* project. — http://eu-datagrid.web.cern.ch/
29. *Almond J., Snelling D.* UNICORE: uniform access to supercomputing as an element of electronic commerce // Future Generation Computer Systems, NH-Elsevier. — 1999. — **15**. — P. 539–548.
30. *EuroGrid*. — http://www.eurogrid.org/
31. *Web* Based Metacomputing / T. Haupt, E. Akarsu, G. Fox, W. Furmanski // Future Generation Computer Systems. — North Holland, 1999. — P. 89–101.
32. *Clark D.* Face-to-Face with Peer-to-Peer Networking // Computer. — Jan. 2001. — **34(**1). — P. 18–21.
33. *Napster*. — http://www.napster.com/
34. *Gnutella*. — http://www.gnutella.co.uk/
35. *Entropia*. — http://entropia.com/
36. *JXTA*. — http://www.jxta.org/
37. *Foster I., Kesselman C., Tuecke S.* The Anatomy of the Grid. — [4], 2003. — P. 171–198.
38. *Fox G.C.* From Computational Science to Internetics: Integration of Science with Computer Science / Eds. R.F. Boisvert, E. Houstis, Computational Science, Mathematics and Software. — West Lafayette, Indiana: Purdue University Press. — 2002. — P. 217–236.
39. *IBM* Autonomic Computing. — http://www.research.ibm.com/autonomic/
40. *NASA* Information Power Grid. — http://www.ipg.nasa.gov/
41. *Web Services* Description Language (WSDL). V. 1.1. W3C Note 15. March 2001. — http://www.w3.org/TR/wsdl
42. *UDDI*. — http://www.uddi.org/
43. *Web Services* Flow Language (WSFL). V. 1.0. — http://www-4.ibm.com/software/solutions/ /Webservices/pdf/WSFL.pdf
44. *Web Services* for Business Process Design. — http://www.gotdotnet.com/team/xml_wsspecs/ /xlang-c/default.htm
45. *Web Services* Modelling Framework. — http://www.cs.vu.nl/~dieter/wese/
46. *DAML-S*: Web Service Description for the Semantic Web // The First Intern. Semantic Web Conf. (ISWC). — June, 2002. — P. 348–363.
47. *Jennings N. R.* An agent-based approach for building complex software systems. Comm. ACM. — 2002. — **44**(4). — P. 35–41.
48. *Wooldridge M., Jennings N.R.* Intelligent Agents: Theory and Practice // Knowledge Engineering Review. — 1995. — **10**(2). — P. 48–58.
49. Jennings N. R., Faratin P., Lomuscio A. R., Parsons S., Sierra C., M. Wooldridge Automated Negotiation: Prospects, Methods and

Challenges / J. of Group Decision and Negotiation. — 2001. — **10**(2). — P. 199–215.
50. *Web-based* Distributed Authoring and Versioning. — http://www.Webdav.org/
51. *W3C Semantic* Web Activity Statement. — http://www.w3.org/2001/sw/Activity
52. *Hendler J., McGuinness D.* The DARPA Agent Markup Language // IEEE Intelligent Systems. — 2000. — **15(**6). — P. 72–73.
53. *De Roure D., Jennings N. R., Shadbolt N. R.* The Semantic Grid: A Future e-Science Infrastructure. — [4], 2003. — P. 437–470.
54. *Coen M.A.* A Prototype Intelligent Environment / Eds. N. Streitz, S. Konomi, H-J. Burkhardt // Cooperative Buildings — Integrating Information, Organisation and Architecture. — Springer-Verlag. — 1998. — P. 156–167.
55. *Access* Grid. — http://www.accessgrid.org/

Получено 24.12.04

Об авторах

Дорошенко Анатолий Ефимович,
д-р физ.-мат. наук, профессор,
зам. директора по научн. работе

Алистратов Олег Валентинович,
научный сотрудник

Турчак Юрий Марьянович,
аспирант

Розенблат Александр Петрович,
инженер-программист

Рухлис Константин Александрович,
аспирант

Место работы авторов:
Институт программных систем НАН Украины,
Просп. Академика Глушкова, 40,
Киев-187, 03680, Украина
Тел. 266 1538
E-mail: dor@isofts.kiev.ua