

**Московский Государственный Университет
имени М.В. Ломоносова**

Научно-исследовательский институт ядерной физики
имени Д.В. Скобельцына

А.П. Демичев, В.А. Ильин, А.П. Крюков

Введение в грид-технологии

Препринт НИИЯФ МГУ - 2007 - 11/832

Москва 2007

А.П. Демичев, В.А. Ильин, А.П. Крюков
demichev@theory.sinp.msu.ru

Введение в грид-технологии

Препринт НИИЯФ МГУ - 2007 - 11/832

Аннотация

В статье дается введение в распределенные вычисления (distributed computing) — способ решения трудоемких задач (вычислительных и задач обработки данных) с привлечением большого числа территориально удаленных друг от друга вычислительных ресурсов, а также ресурсов хранения и передачи данных. Вообще говоря, термин "распределенные вычисления" объединяет большое число различных подходов и конкретных технологий для решения такого рода задач. Основное внимание в данной статье уделяется одному из таких подходов, а именно, технологии создания и использования больших грид-систем.

Работа выполнена при поддержке Европейской Комиссии (в рамках проекта EGEE), фонда РФФИ (грант № 05-07-90292) и Федерального агентства по науке и инновациям (грант NS1685.2003.2).

A.P.Demichev, V.A.Ilyin, A.P.Kryukov

Introduction to Grid Technology

Препринт НИИЯФ МГУ - 2007 - 11/832

Аннотация

In this paper we give an introduction into distributed computing, a way of solving complicated computational and data processing problems by means of using large number of computing and data storage resources which are remote from each other. In general, the term "distributed computing" includes large number of various approaches and specific technologies for solving such problems. The main attention in this paper is devoted to a specific approach, namely, to technologies of creation and exploitation of large grid systems.

This work was supported by European Commission (in the framework of EGEE project), Russian Foundation for Basic Research (grant No. 05-07-90292) and The Russian Federal Agency for Science and Innovations (grant No. NS1685.2003.2).

©А.П. Демичев, В.А. Ильин, А.П. Крюков, 2007

©НИИЯФ МГУ, 2007, <http://www.sinp.msu.ru>

Содержание

1 ЧТО ТАКОЕ ГРИД, ЗАЧЕМ НУЖЕН, КАК ОН ПОЯВИЛСЯ И КАК УСТРОЕН.....	5
1.1 Что такое грид.....	5
1.1.1 Терминология и определение грида.....	5
1.2 Зачем нужен грид.....	7
1.2.1 Общие задачи грида	7
1.2.1.1 Обеспечение распределенных вычислений и обработки данных (удаленный доступ к вычислительным ресурсам).....	8
1.2.1.2 Повышение эффективности компьютерных ресурсов.....	8
1.2.2 Типы грид-систем с точки зрения решаемых задач.....	9
1.2.3 Задачи грида и задачи суперкомпьютеров (сходство и различие).....	9
1.3 Как появился грид.....	11
1.3.1 От WWW к гриду (через веб-сервисы)	11
1.3.2 Еще немного истории: пионеры грид-движения.....	12
1.3.3 Кто занимается стандартизацией веб/грид-технологий.....	13
1.4 Как грид устроен.....	14
1.4.1 Каким он должен быть.....	14
1.4.2 Что является, и что не является гридом: грид, кластеры и распределенные вычисления.....	14
1.4.3 Общие принципы архитектуры грид-систем.....	17
1.4.3.1 Структура (стек) протоколов глобального грида	17
1.4.3.1.1 Аппаратный уровень: управление локальными ресурсами	18
1.4.3.1.2 Связывающий уровень: коммуникации и безопасность	18
1.4.3.1.3 Ресурсный уровень: совместное использование ресурсов	18
1.4.3.1.4 Коллективный уровень: координация ресурсов	19
1.4.3.1.5 Прикладной уровень: запуск приложений в грид-среде	19
1.4.3.2 Архитектура сервисов распределенных систем и технологии ее реализации.....	19
1.4.3.2.1 Сервисно-ориентированная архитектура (SOA) распределенных систем.....	19
1.4.3.2.2 Принцип слабой связи.....	21
1.4.3.2.3 Понятие состояния сервисов и сервисы без состояний.....	22
1.4.3.2.4 Веб-сервисы.....	23
1.4.3.2.5 Веб-сервисы и SOA.....	24
1.4.3.2.6 Веб-сервисы и виртуализация.....	24
1.4.3.2.7 Сервисно-ориентированный грид.....	25
1.4.3.2.8 Open Grid Services Architecture (OGSA).....	26
1.4.3.2.9 Грид-спецификации WSRF (Web Services Resource Framework) и WS-Notification.....	29
1.4.3.2.10 Другие грид-стандарты.....	30
1.4.3.2.11 Сервисно-ориентированные и объектно-ориентированные системы: сходство и различия, преимущества и недостатки.....	30
1.4.3.2.12 Разработка систем на основе SOA.....	32
1.5 ЗАКЛЮЧЕНИЕ К ПЕРВОЙ ЧАСТИ.....	33
2 ОСНОВНЫЕ ФУНКЦИОНАЛЬНЫЕ ПОДСИСТЕМЫ ГЛОБАЛЬНОГО ГРИДА.....	34
2.1 Краткое описание функционального назначения базовых подсистем.....	36
2.2 Подсистема управления загрузкой.....	37
2.2.1 Компоненты подсистемы управления загрузкой.....	37
2.2.2 Взаимодействие подсистемы управления загрузкой с другими грид-службами.....	39
2.2.3 Вычислительный элемент.....	40
2.3 Подсистема управления данными.....	41
2.3.1 Наименование файлов в глобальном гриде.....	41
2.3.2 Ресурсы хранения данных.....	43
2.3.3 Каталоги.....	44
2.4 Подсистема информационного обслуживания и мониторинга грида.....	45
2.4.1 Архитектура и реализация	45
2.4.2 Структура поставщиков и потребителей подсистемы информационного обслуживания и мониторинга на основе R-GMA.....	47
2.4.3 Период хранения кортежей.....	48
2.5 Подсистема безопасности и контроля прав доступа.....	48
2.5.1 Некоторые термины и общие принципы алгоритмов шифрования.....	49
2.5.2 Идентификация пользователей и узлов грида.....	50
2.5.3 Делегирование прав и использование доверенностей.....	51
2.5.4 Сервис управления виртуальными организациями и авторизация пользователей.....	52
2.6 Подсистема протоколирования.....	53
2.6.1 Типы запросов к подсистеме протоколирования.....	53

2.6.2	Безопасность и управление доступом к информации подсистемы протоколирования.....	53
2.6.3	Источники извещений о событиях.....	54
2.7	ПОДСИСТЕМА УЧЕТА.....	54
2.7.1	Подсистема учета ППО gLite.....	55
3	РЕАЛИЗАЦИЯ ГРИД-ТЕХНОЛОГИЙ В ПРОЕКТЕ EGEE/LCG/RDIG.....	57
3.1	ОСНОВНЫЕ ЦЕЛИ И СТРУКТУРА ПРОЕКТА EGEE	57
3.1.1	Направления работ по проекту	58
3.1.2	Критерии успеха проекта	59
3.2	КОМПОНЕНТЫ ГРИДА EGEE.....	59
3.2.1	Технологии и стандарты: промежуточное программное обеспечение грида EGEE.....	59
3.3	ИНФРАСТРУКТУРА.....	60
3.3.1	Базовая компьютерная сеть: GEANT.....	62
3.3.2	Инфраструктура безопасной работы в грид-среде.....	63
3.3.3	Основная организационная единица для пользователей грида – виртуальная организация.....	63
3.4	RDIG: РОССИЙСКИЕ УЧЕНЫЕ В ПРОЕКТЕ EGEE.....	63
3.4.1	Региональные российские виртуальные организации.....	65
3.5	ГРИД-СРЕДА EGEE ГЛАЗАМИ ПОЛЬЗОВАТЕЛЕЙ И ПРОВАЙДЕРОВ РЕСУРСОВ	65
3.5.1	Грид-среда глазами провайдера ресурсов.....	65
3.5.2	Грид-среда глазами пользователя.....	66
3.6	ПРИКЛАДНЫЕ ЗАДАЧИ В ПРОЕКТЕ EGEE.....	67
3.6.1	Физика высоких энергий: проект LCG.....	67
3.6.2	Астрофизические приложения.....	68
3.6.3	Приложения в области ядерного синтеза.....	69
3.6.4	Решение задач биомедицины с помощью грид-среды EGEE.....	69
3.6.4.1	Drug Discovery: борьба с массовыми болезнями с помощью грида.....	71
3.6.5	Приложения в области вычислительной химии.....	72
3.6.6	Приложения для наук о Земле и геофизики.....	73
3.6.6.1	Грид для предсказания извержений вулканов.....	74
3.6.7	Финансы и мультимедиа	74
3.6.8	Грид-приложение в области археологии.....	74
4	ЗАКЛЮЧЕНИЕ.....	75
	КРАТКИЙ ГЛОССАРИЙ ТЕРМИНОВ, СВЯЗАННЫХ С ВЕБ/ГРИД-ТЕХНОЛОГИЯМИ.....	77
	СПИСОК ЛИТЕРАТУРЫ.....	83

1 Что такое грид, зачем нужен, как он появился и как устроен

1.1 Что такое грид

1.1.1 Терминология и определение грида

В широком смысле, предметом этой статьи являются распределенные вычисления (distributed computing) — способ решения трудоемких задач (вычислительных и задач обработки данных) с привлечением большого числа исполнителей (вычислительных ресурсов, а также ресурсов хранения и передачи данных), работающих одновременно над разными частями задачи. Вообще говоря, этот термин объединяет большое число различных подходов и конкретных технологий для решения такого рода задач. Мы, однако, сосредоточимся на одном из подходов, а именно, на технологии создания и использования больших грид-систем.

Для начала определим более точно – что такое грид и чем грид-технологии отличаются от других типов распределенных вычислений. В основополагающей книге [1] пионеры грид-технологий – Ян Фостер (Ian Foster) из Аргоннской Национальной лаборатории и Чикагского университета и Карл Кессельман из Института информационных наук Университета Южной Калифорнии (США) дают следующее определение: "Грид - согласованная, открытая и стандартизованная среда, которая обеспечивает гибкое, безопасное, скоординированное разделение (общий доступ) ресурсов в рамках виртуальной организации".

Прежде, чем более подробно объяснить смысл этого определения – поясним происхождение самого названия такой среды, которое несет вполне определенную смысловую нагрузку. Термин грид (в смысле одной из технологий распределенных вычислений - возник в середине 90-х годов) происходит от английского слова «grid» и его буквальным переводом на русский язык является «решетка». В русскоязычной литературе зачастую и используется этот буквальный перевод. В частности, мы использовали его в названии нашей статьи – для того, чтобы подчеркнуть процесс эволюции современных компьютерных технологий. Однако надо понимать, что такой буквальный перевод не вполне отражает смысл термина. Дело в том, что англоязычный термин grid произошел от «power grid», что соответствует русскому «электросеть» или «энергосистема». В этом и заключается смысл этого названия: подобно тому, как при использовании энергосистем мы не интересуемся – какой конкретный электрогенератор выработал ток, который мы потребляем (а просто платим за потребленное электричество энергетической компании, с которой мы заключили договор), при использовании компьютерного грида мы можем не заботиться о том - какой конкретно компьютер (или устройство хранения/передачи данных) в грид-системе выполнил нашу задачу (при этом, возможно, придется тем или иным способом платить соответствующему провайдеру за использованные ресурсы).

Эту аналогию можно продолжить. Современный статус компьютерных инфраструктур можно сравнить с состоянием электрических систем в 1910 году. Тогда необходимость для каждого пользователя иметь свой генератор тормозила развитие отрасли. Революционным шагом было возникновение электросетей, создание технологий передачи и распределения электроэнергии, создание *стандартизованной службы* универсального и гарантированного доступа к электроэнергии. В результате не только резко повысилась эффективность использования ресурсов, но и стали возможны принципиально новые направления развития. Теперь, купив электрический чайник, мы просто втыкаем вилку в

электрическую розетку и начинаем им пользоваться: электросеть избавляет нас от необходимости вместе с чайником покупать и источник энергии – электрогенератор. Аналогично, грид призван обеспечить возможность делать компьютерные вычисления «по требованию» - просто подключившись к «решетке» вычислительных ресурсов.

Также как и электрические сети, грид это соединение *технологии, инфраструктуры и стандартов*. Технология – это специальное программное обеспечение, которое позволяет организациям или частным лицам предоставлять ресурсы (компьютеры, хранилища данных, сети и другие) в общее пользование, а потребителям – использовать их, когда необходимо. Инфраструктура состоит из аппаратных средств и служб (на основе людских и программных ресурсов), которые должны быть организованы, и постоянно поддерживаться для того, чтобы ресурсы могли совместно использоваться. Наконец, стандарты должны определять формат и протоколы обмена сообщениями, как между службами, так и между службами и пользователями, а также правила работы грида. Важность этих составляющих грида можно пояснить, опять обращаясь к аналогии с энергосетями, в которых определяющую роль играют:

- *технологии* передачи и (пере)распределения электроэнергии на большие расстояния;
- *инфраструктура* - аппаратные средства (повышающие/понижающие подстанции, линии электропередач, другое электрооборудование, позволяющее электростанциям предоставлять энергию потребителям) и соответствующие службы (ремонта, контроля и т.д.);
- *стандарты* – параметры электрического тока (напряжение в сети, частота), типы и размеры вилок/розеток, позволяющие без проблем подключать любой электроприбор к единой сети.

Итак, в основе грид-систем лежит обеспечение стабильной работы набора служб на основе общепринятых открытых стандартов и управляющего программного обеспечения (промежуточного программного обеспечения (ППО), в английском языке используется термин *middleware*) для обеспечения надежного, унифицированного доступа к географически распределенным информационным и вычислительным ресурсам, включающим отдельные компьютеры, кластеры и суперкомпьютерные центры, хранилища информации и т.д. Создание таких систем стало возможным благодаря впечатляющим успехам, прежде всего, в четырех направлениях:

- повышению производительности микропроцессоров массового производства - современный персональный компьютер сравним по производительности с суперкомпьютерами десятилетней давности;
- появлению быстрых линий связи - сейчас осуществляется перевод основных магистралей на уровень нескольких Гигабит/сек;
- глобализации обмена информацией (Интернет/Веб);
- развитию методов метакомпьютинга - научной дисциплины по организации массовых и распределенных вычислительных процессов.

Поскольку грид-инфраструктура основана на предоставлении ресурсов в общее пользование, с одной стороны, и на использовании публично доступных ресурсов, с другой, одним из ее ключевых понятий (вспомните общее определение грида в начале этого раздела) является виртуальная организация (ВО). Подробнее это понятие обсуждается в разделе 3.3. Здесь лишь отметим, что ВО является сообществом людей, которые совместно используют грид-ресурсы в соответствии с согласованными между ними и собственниками ресурсов правилами. Фактически, виртуальная организация заключает договор с владельцами ресурсов об их использовании и является представителем членов данной ВО во взаимоотношениях с владельцами ресурсов. В существующих грид-системах виртуальные организации образуют специалисты из некоторой прикладной области, которые объединяются для достижения общей цели.

Любая ВО имеет доступ к определенному набору ресурсов, которые предоставляются зарегистрированным в ней пользователям. С другой стороны, каждый ресурс может одновременно предоставляться нескольким ВО. Каждая ВО самостоятельно устанавливает правила работы для своих участников, исходя из соблюдения баланса между потребностями пользователей и наличным объемом ресурсов, поэтому пользователь должен обосновать свое желание работать с грид-системой и получить согласие управляющих органов ВО.

1.2 Зачем нужен грид

1.2.1 Общие задачи грида

Концепция грида появилась не как абстрактная идея, а как ответ на появляющиеся потребности в крупных информационно-вычислительных ресурсах, динамически выделяемых для решения громоздких задач, в научной, индустриальной, административной и коммерческой областях деятельности. Создание грид-среды подразумевает распределение вычислительных ресурсов по территориально разделенным сайтам, на которых установлено специализированное программное обеспечение для того, чтобы распределять задания по сайтам и принимать их там, возвращать результаты пользователю, контролировать права пользователей на доступ к тем или иным ресурсам, осуществлять мониторинг ресурсов, и так далее. Общедоступные ресурсы на основе сайта могут включать вычислительные узлы и/или узлы хранения и передачи данных, собственно данные, прикладное программное обеспечение.

Вычислительные ресурсы предоставляют пользователю грид-системы (точнее говоря, задаче пользователя) процессорные мощности. Вычислительными ресурсами могут быть как кластеры, так и отдельные рабочие станции. При всем разнообразии архитектур любая вычислительная система может рассматриваться как потенциальный вычислительный ресурс грид-системы. Необходимым условием для этого является наличие ППО, реализующего стандартный внешний интерфейс с ресурсом и позволяющего сделать ресурс доступным для грид-системы. Основной характеристикой вычислительного ресурса является производительность (см., например, <http://www.spec.org>).

Ресурсы хранения также используют ППО, реализующее унифицированный интерфейс управления и передачи данных. Как и в случае вычислительных ресурсов, физическая архитектура ресурса памяти не принципиальна для грид-системы, будь то жесткий диск на рабочей станции или система массового хранения данных на сотни терабайт. Основной характеристикой ресурсов хранения данных является их объем. В настоящее время характерный объем ресурсов хранения измеряется в Терабайтах (Тб).

Информационные ресурсы и каталоги являются особым видом ресурсов хранения данных. Они служат для хранения и предоставления метаданных и информации о других ресурсах грид-системы. Информационные ресурсы позволяют структурировано хранить огромный объем информации о текущем состоянии грид-системы и эффективно выполнять задачи поиска ресурсов.

Сетевой ресурс является связующим звеном между распределенными ресурсами грид-системы. Основной характеристикой сетевого ресурса является скорость передачи данных.

Важнейшим является междисциплинарный характер работ по развитию грид-вычислений – уже сегодня эти технологии применяются в самых разных прикладных областях. В мире возникли сотни грид-форумов (например, [2]) и проектов [3] - в физике высоких энергий, космофизике, микробиологии, экологии, метеорологии, различных инженерных приложениях (например, в самолетостроении).

Основными общими задачами грида являются:

- создание из серийно выпускаемого оборудования широкомасштабных распределенных вычислительных систем и систем обработки, комплексного анализа и мониторинга данных, источники которых также могут быть (глобально) распределены;
- повышение эффективности вычислительной техники путем предоставления в грид временно простаивающих ресурсов.

Приоритет той или иной общей задачи, которая решается с помощью грида, определяется типом грида и характером прикладных областей, в которых он используется.

1.2.1.1 Обеспечение распределенных вычислений и обработки данных (удаленный доступ к вычислительным ресурсам)

Как уже отмечалось, компьютеры совершенствуются невероятно быстро: процессорная мощность удваивается приблизительно каждые 18 месяцев. Но и такой прогресс не всегда обеспечивает потребности ученых и инженеров - часто для вычислений недостаточно одного компьютера или даже кластера персональных компьютеров. В результате достижение научных или инженерных целей может оказаться чрезвычайно сложной, очень дорогой, а иногда совершенно невыполнимой задачей. Если же на время выполнения задачи или проекта есть возможность использовать ресурсы множества персональных компьютеров, рабочих станций, кластеров, может быть даже и суперкомпьютеров, а также хранилищ данных, размещенных в разных точках мира и принадлежащих разным людям и учреждениям, то проблема может стать разрешимой. Такую возможность – распределения отдельных частей большой задачи по географически удаленным ресурсам (если сам характер задачи позволяет разделить ее на части) и предоставляет грид-среда.

Особенно эффективным такой подход оказывается, когда в рамках какого-либо проекта (исследования или прикладной проблемы) надо выполнить огромный поток или набор однотипных задач. Типичным примером может служить обработка большого объема данных.

1.2.1.2 Повышение эффективности компьютерных ресурсов

Другая общая задача, которую решают системы распределенных вычислений, является вполне традиционной для любого производственного процесса: это повышение эффективности использования оборудования (средств производства). Эта задача становится особенно актуальной, когда производительность средств производства становится выше, чем требуется для удовлетворения локальных потребностей. Например, пока землю пахали с помощью лошади и сохи, каждый крестьянин имел свои орудия труда (производительность была так низка, что делиться с другими было просто невозможно: необходимый объем работ едва можно было выполнить за сезон). Трактор и плуг, напротив, очень часто используются совместно (колхозы, МТС, объединения фермеров и т.п.) – это возможно благодаря их высокой производительности и экономически выгодно. Аналогично и в случае компьютеров: пока их производительность была невелика, они едва справлялись с локальными задачами в однопользовательском режиме. Современные же мощные ПК зачастую реально используются не более, чем на несколько процентов своей мощности, и естественной становится задача предоставления неиспользуемых ресурсов для общественных нужд. Аналогичная ситуация характерна и для компьютерных центров, поскольку большие вычислительные задачи появляются у пользователей не постоянно, то есть потребность в выполнении вычислительной работы имеет пиковый по времени характер. Грид-технологии позволяют объединять эти мощности в единую, хотя и географически распределенную вычислительную среду. В результате пользователь получает возможность запуска своих задач на таком «глобальном

кластере», имеющем мощность, существенно превосходящую его собственные ресурсы. При этом и его компьютер (или компьютеры) может быть включен в состав этого глобального кластера, и будет выполнять задачи других пользователей. Результат же вычислений будет получен существенно быстрее, чем при последовательном запуске задач на своем персональном компьютере. Следует подчеркнуть, что временной эффект будет связан не только с пиковым характером появления задач у данного пользователя, что может иметь периодичность в дни и даже месяцы. Эффект может быть получен и просто из-за неравномерного запуска задач в течение дня, в силу чего пики загрузки не совпадают для географически разделенных по временным поясам районов (такой эффект очень хорошо известен и в электрических сетях).

1.2.2 Типы грид-систем с точки зрения решаемых задач

Анализируя существующие проекты по построению грид-систем можно сделать вывод о трех направлениях развития грид-технологии:

1. вычислительный грид (Computational Grid),
2. грид для интенсивной обработки данных(Data Grid),
3. семантический Грид для оперирования данными из различных баз данных (Semantic Grid).

Целью первого направления является достижение максимальной скорости вычислений за счет глобального распределения этих вычислений между тысячами компьютеров, а также, возможно, серверами и суперкомпьютерами.

Целью второго направления является обработка огромных объемов данных относительно несложными программами. Поэтому вычислительные ресурсы грид-инфраструктуры в этом случае зачастую представляют собой кластеры персональных компьютеров. А вот доставка данных для обработки и пересылка результатов в этом случае представляют собой достаточно сложную задачу. Одним из крупнейших проектов, целью которого является создание грид-системы для обработки научных данных, является проект EGEE (Enabling Grids for E-sciencE) [13]. О нем мы будем подробно рассказывать в части 3 и в контексте этого рассказа обсудим вопросы, связанные с хранением и (своевременной) передачей больших объемов данных.

Грид-системы третьего направления - семантические - предоставляют инфраструктуру для выполнения вычислительных задач на основе распределенного мета-информационного окружения, позволяющего оперировать данными из разнотипных баз, различных форматов, представляя результат в формате, определяемом приложением. В этом обзоре мы совсем не будем осуждать этот тип грид-систем и просто отсылаем читателя к соответствующему веб-сайту [27].

1.2.3 Задачи грида и задачи суперкомпьютеров (сходство и различие)

Не все проблемы лучше всего решать, используя распределенные кластеры на основе грид-технологий. Суперкомпьютеры незаменимы для некоторых научных проблем, типа составления прогноза погоды, когда множество процессоров должны часто общаться друг с другом. Очевидно, что такое частое общение невозможно обеспечить для географически распределенных и, возможно, аппаратно-неоднородных ресурсов в грид-среде. Другими словами, грид не слишком подходит для параллельных вычислений с интенсивным межпроцессорным обменом [4].

Поясним чуть более подробно почему не следует смешивать грид-технологии с технологией параллельных вычислений. Основными препятствиями для осуществления нетривиальных параллельных вычислений в грид-среде является нестабильность, плохая предсказуемость времени отклика на запрос. Причем это связано не только с тем, что в компьютерных сетях информационные пакеты проходят через множество сетевых

устройств, но и с различиями в протоколах связи используемых во "внешних" компьютерных сетях и для межпроцессорного обмена внутри суперкомпьютеров. Это не позволяет эффективно организовать параллельные вычисления с интенсивным обменом информацией между процессорами, выполняющими отдельные подзадачи, в грид-среде.

Грид-технология не является технологией параллельных вычислений, она предназначена для удаленного запуска отдельных задач на территориально распределенные ресурсы. Поэтому если громоздкая задача, которую необходимо решить, может быть разбита на большое количество маленьких, *независимых* (не обменивающихся никакими данными) частей, - грид-технология оказывается особенно эффективным и относительно дешевым решением. Напротив, суперкомпьютеры оказываются для таких вычислений неоправданно дорогим и неэффективным решением. В англоязычной литературе такие прикладные задачи иногда называют «bag-of-tasks» - сумка/мешок задач: вычисления для каждой выполняются независимо, а в конце пользователь или программное обеспечение просто должны соединить результаты индивидуальных вычислений. Типичными примерами таких задач являются:

- массовая обработка потоков экспериментальных данных большого объема (зачастую изучаемое явление можно разделить на отдельные независимые события и экспериментальные результаты по каждому событию обрабатывать независимо от других);
- визуализация больших наборов данных (отдельные области визуального представления обрабатываются независимо, а потом «склеиваются»);
- сложные бизнес-приложения с большими объемами вычислений (разбиение на части зависит от конкретного характера задачи).

Ради справедливости, надо отметить, что ПО промежуточного слоя нового поколения для грид-систем (в частности, gLite, о котором мы еще будем говорить ниже) позволяет управлять некоторым классом нетривиальных параллельных вычислений – таких, зависимость подзадач в которых может быть представлена направленным графом без циклов (Direct Acyclic Graph (DAG); на Рис. 1 приведен пример такого графа). Но насколько эффективной на практике окажется эта возможность – еще предстоит выяснить.

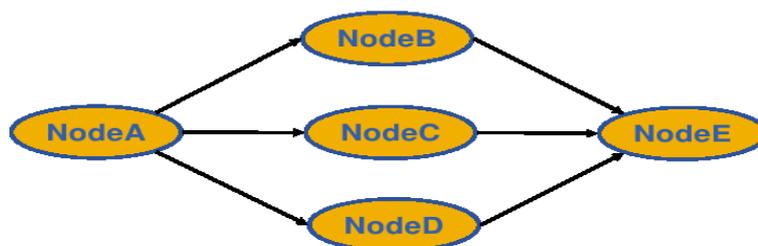


Рис. 1 Пример направленного графа без циклов

Необходимо отметить, что существуют гибридные проекты, целью которых является достижение максимальной скорости вычислений за счет глобального распределения этих вычислений между суперкомпьютерами – при этом грид координирует использование различных суперкомпьютеров, а собственно нетривиальное распараллеливание происходит внутри суперкомпьютера. Проект DEISA [31] может служить примером этого направления, в котором предпринимается попытка объединения суперкомпьютерных центров.

1.3 Как появился грид

1.3.1 От WWW к гриду (через веб-сервисы)

До сих пор мы говорили об аналогии с энергосетями и соотношении задач для грида и суперкомпьютеров. Но, естественно, идея грид-вычислений не могла бы возникнуть без предварительного развития Интернет-технологий.

Интернет, Всемирная паутина (World Wide Web (WWW), веб) и грид – связанные между собой, но различные технологии. Интернет это глобальная система сетей, соединяющая множество компьютеров и локальных (сравнительно небольших) сетей и позволяющая им взаимодействовать друг с другом. Веб это способ доступа к информации находящейся на удаленном, но включенном в Интернет компьютере. Грид – способ совместного использования ресурсов, распределенных по разным, географически удаленным друг от друга, точкам планеты.

Когда в 1991 Тим Бернерс-Ли (Tim Berners-Lee) разрабатывал в Европейском Центре Ядерных Исследований (ЦЕРН, расположен в Женеве) основы технологии Всемирной паутины, он имел побудительные мотивы, сходные с основоположниками грид-компьютинга: в случае веба – потребность в обеспечении универсального и стандартизированного доступа к информационным ресурсам; в случае грида - потребность в обеспечении универсального и стандартизированного доступа к вычислительным ресурсам и ресурсам хранения данных. Интересно, что и в случае грида одним из основных центров разработки новой технологии тоже является ЦЕРН – он вносит определяющий вклад в создание крупнейшего научного грида в рамках проекта “Enabling Grid for EsienceE” (ниже мы расскажем об этом подробнее).

Конечно, с технической точки зрения статические веб-страницы (а такие только и существовали в начале эпохи WWW) и грид имеют не очень много общего – пожалуй, только то, что основой для передачи информации в обоих случаях являются интернетовские транспортные протоколы TCP/IP.

Однако сейчас многие веб-сайты предоставляют динамические веб-страницы: содержание таких страниц может меняться в зависимости от запросов, которые делают пользователи (например, через веб-формы). Это уже несколько ближе к гриду - удаленные компьютеры предоставляют некоторые услуги (например, перевод фрагмента текста с одного языка на другой: это ни что иное, как частный случай обработки данных – прямая аналогия с задачами грида). Намного интереснее и ближе к грид-технологиям то, что зачастую происходит «позади» динамических веб-страниц. Например, вы делаете заказ какого-либо товара на (динамической) странице интернет-магазина. После этого начинается обработка заказа: информация о заказе и клиенте заносится в базу данных, через банковскую систему проверяется действительность кредитной карты клиента, запрос на товар передается на склад магазина или поставщику, и так далее. Если в компьютерном отношении этот магазин достаточно «продвинутый», эти действия выполняются не людьми, а специальными компонентами программного обеспечения – веб-сервисами, которые взаимодействуют друг с другом по стандартизованным протоколам. Точное определение веб-сервисов мы откладываем до раздела 1.4. Отметим, что в русскоязычной литературе и в случае WWW, и в случае грида используют два варианта перевода англоязычных терминов web/grid services: веб-/грид-сервисы и веб-/грид-службы. Мы будем также использовать оба этих варианта - как полностью эквивалентные.

Итак, веб-страницы служат для обмена информацией между людьми, веб-сервисы – для взаимодействия компьютеров (точнее – прикладных программ на различных компьютерах) друг с другом. Отсюда – один шаг до создания системы (грид-)служб для запуска заданий на удаленных ресурсах, обработки и передачи данных, их (задач и данных) мониторинга и сбора результатов. С общей функциональной точки зрения, от

веб-сервисов грид отличается только тем, что каждая система веб-сервисов настроена на решение узкого набора конкретных задач (например, обработки заказа в интернет-магазине), а грид – на решение широкого круга вычислительных задач и задач обработки и передачи данных на удаленных ресурсах. Но программные компоненты грид-среды, которые обеспечивают распределение вычислительных заданий, контроль их выполнения, передачу данных и т.д., могут быть созданы на основе веб-сервисных технологий – правда, с некоторыми расширениями и дополнительными стандартами. Более подробно мы рассмотрим этот вопрос в разделе 1.4.

Необходимо отметить, что хотя в качестве предпосылки к появления грида мы упомянули только WWW, в действительности были и другие достижения 90-х годов, ставшие как важными вехами развития компьютерных технологий в целом, так важными предпосылками для появления грид-технологий. Среди них нужно упомянуть Linux (1991 год) и кластеры типа Beowulf (1994 год), которые инициировали широкое распространение *централизованных* систем с разделяемым доступом. Следующим естественным шагом – на основе Интернета и WWW – стало развитие децентрализованных распределенных систем.

1.3.2 Еще немного истории: пионеры грид-движения

Основной теоретической предшественницей современных грид-проектов считается инициатива Metacomputing, предложенная в середине 80-х годов исследователями из Национального центра суперкомпьютерных приложений США (см., например, [30]). Ее главная идея состояла в объединении нескольких суперкомпьютеров для достижения большей производительности. Одной из первых инфраструктур, реализующих эту идею, стала в 1995 году Wide Area Year (I-WAY). Йан Фостер и Карл Кессельман, участвовавшие в разработке проекта, в том же году опубликовали первые материалы, а в 1997 году провели первый семинар на эту тему (Построение вычислительного грида - Building a Computational Grid). Это и было рождением концепции грида. Позднее они стали редакторами знаменитых книг [1], заложивших основы построения грид-систем.

Необходимо отметить, однако, что Metacomputing была не единственной инициативой такого рода. Среди схожих по концепции проектов конца 80-х – начала 90-х можно отметить:

- Condor (университета штата Висконсин, США);
- CODINE (Computing for Distributed Network Environments, немецкая компания Genias Software - позднее переименована в Gridware, в 1992 году куплена корпорацией Sun Microsystems);
- Legion (университет штата Вирджиния, позднее рабочая группа проекта выделилась в компанию Avaki).

В 1998-2002 годах были разработаны общие основы создания сетевых структур для использования географически разнесенных и находящихся в разном подчинении вычислительных мощностей. Эти результаты были суммированы в [1], а термин «грид» стал общепринятым. Вскоре группа исследователей из Чикагского университета, возглавляемая Йаном Фостером, поддерживаемая IBM и Национальной арагонской лабораторией, разработала Открытую архитектуру грид-сервисов (Open Grid Services Architecture, OGSA), основанную на специализированных грид-сервисах.

Для создания базовой (reference) модели грида разработчиками OGSA была организована рабочая группа Open Grid Service Infrastructure (Инфраструктура открытого грида, OGSi). В 2003 году она выпустила спецификацию OGSi, в которой определены механизмы создания, управления и обмена данными между грид-сервисами. Однако, из-за ряда недостатков (в частности, чрезвычайной громоздкости) эта спецификация не была хорошо

воспринята ни практиками-разработчиками грид-систем, ни Веб-сообществом. Поэтому естественным стало появление набора спецификаций Web Service Resource Framework (WSRF), где еще точнее отражено сходство грид- и веб-сервисов. Важно, что новый стандарт влияет не на всю концепцию OGSA — на верхнем уровне все остается неизменным. Как и веб-сервисы, OGSA/WSRF-сервисы могут быть определены в терминах языка WSDL, что позволяет использовать преимущества известных стандартов SOAP, XML и WS-Security (подробнее об этом – в разделе 1.4).

В 2004 году компании BEA Systems, Microsoft и Tibco опубликовали свою спецификацию для веб-сервисов - WS-Eventing, основанную на принципах публикации/подписки. Она не привязана непосредственно к гриду, а предназначена для описания взаимодействия между веб-сервисами (подробнее – также в разделе 1.4). Цели этой спецификации весьма близки к целям WSRF, но достигаются другими средствами. Почти одновременно компании Akamai, Globus Alliance, Hewlett-Packard, IBM, Sonic Software и все та же Tibco предложили спецификацию WS-Notification, входящую в состав WS-Resource Framework и опять-таки реализующую принцип публикации/подписки.

Наконец весной 2006 года компании HP, IBM, Intel и Microsoft объявили о своем намерении участвовать в разработке общего стандартного набора спецификаций веб-служб и выработали «дорожную карту» общей схемы этих спецификаций [33]. Эта дорожная карта предлагает способ согласования этих двух подходов: семейства спецификаций Web Services Distributed Management (WSDM) (включающих WSRF и WS-Notification; поддерживается IBM, HP и рядом других организаций) и семейство спецификаций WS-Management (включающих WS-Eventing, WS-Transfer и WS-Enumeration; поддерживается Microsoft, Intel и другими). Опубликованная дорожная карта предлагает, чтобы новые разрабатываемые спецификации в окончательном виде включали все базовые концепции, введенные сначала в OGSI и реализованные впоследствии в WSRF/WS-N.

Таким образом, сейчас появилась реальная надежда получить всеобъемлющие стандарты управления системами, базирующимися на веб-службах – в том числе, управления грид-системами.

1.3.3 Кто занимается стандартизацией веб/грид-технологий

До недавнего времени основной организацией по стандартизации грид-технологий являлся Global Grid Forum (GGF). Это общественная организация, включающая более 5 тыс. специалистов, работающих в области распределенных вычислений и технологий грид. Основная цель GGF - содействие развитию, реализации технологий грид и развертыванию приложений путем создания основополагающих документов (технические спецификации, описания опытов создания приложений, руководства и т.п.).

Помимо Global Grid Forum вопросами стандартизации занимался и созданный в 2004 году Enterprise Grid Alliance (EGA) - консорциум ИТ-производителей (EMC, Fujitsu Siemens, HP, Intel, NEC, Network Appliance, Oracle, Sun Microsystems и целый ряд меньших по размеру компаний), заинтересованных в развитии идей грид. Создавая свой альянс, они руководствовались стремлением применять грид-технологии главным в бизнес-приложениях, а не для решения научных или технических задач, требующих большой вычислительной мощности. Поэтому в центре внимания альянса оказались статические grid-структуры, расположенные в одном географическом месте и являющиеся составляющей корпоративного центра обработки данных.

Однако летом 2006 года эти две ведущие грид-организации объявили о своем объединении. Образована новая организация под названием Open Grid Forum (OGF) [2], объединившая эти две ведущие организации для ускорения внедрения грид-технологий во всем мире. Во главе новой организации будет стоять прежний председатель GGF Марк Линеш (Mark Linesch).

OGF унаследует от EGA все достижения этого альянса в области промышленной экспертизы и ориентацию на быстрые практические результаты, а от GGF - богатый опыт открытого сотрудничества в области исследований грид-систем и в разработке стандартов. OGF является открытым форумом для всех ключевых фигур и организаций грид-сообщества с целью обнаружения потенциальных проблем и их решения на основе обсуждений в рабочих группах, а также разработки спецификаций для ускорения внедрения грида.

Как открытая организация по стандартизации, OGF развивает сотрудничество с другими разрабатывающими стандарты организациями по согласованию существующих промышленных стандартов и разработке новых спецификаций. В число таких организаций входят: Organization for the Advancement of Structured Information Standards (OASIS, www.oasis.org), World Wide Web Consortium (W3C, www.w3c.org), Distributed Management Task Force (DMTF, www.dmtf.org), Web Services Interoperability Organization (WS-I, www.ws-i.org), Internet2 (www.internet2.edu) и Liberty Alliance (www.projectliberty.org).

1.4 Как грид устроен

1.4.1 Каким он должен быть

Правильно разработанная и хорошо реализованная грид-среда характеризуется следующими основными функциональными возможностями [30]:

- доступ к вычислительным ресурсам, данным, устройствам, измерительным инструментам должен быть простым, прозрачным, удаленным, и безопасным;
- доступ должен быть виртуальным (нужен доступ не к серверам, а к сервисам, поставляющим данные или вычислительные ресурсы — причем без необходимости знания аппаратной структуры, обеспечивающей эти сервисы);
- доступ должен осуществляться по требованию (с заданным качеством), а ресурсы должны предоставляться тогда, когда в них возникает нужда;
- доступ должен быть распределенным, обеспечивая возможность совместной коллективной работы виртуальных команд;
- доступ должен быть устойчив к сбоям, а при выходе из строя серверов приложения должны автоматически мигрировать на резервные серверы;
- доступ должен обеспечивать возможность работы в гетерогенной среде – с различными платформами.

Необходимо отметить, что не все из этих требований в должной мере реализованы в настоящее время.

1.4.2 Что является, и что не является гридом: грид, кластеры и распределенные вычисления

Мы выяснили - в чем сходство и различия задач грида и суперкомпьютеров, технологий Интернета, WWW и грида. Теперь давайте обсудим - как соотносится термин «грид» с терминами «кластер» и «распределенные вычисления». Дело в том, что термин «грид» зачастую используется в разных значениях. Особенно сильно, такая тенденция проявилась после 2002 года, когда сотрудниками компании Sun Microsystems, была опубликована привлекающая большое внимание статья [32], в которой термин «грид» использовался для кластерных конфигураций разных масштабов. После этого началась, - и по сей день продолжается - дискуссия о том, что такое кластер, а что — грид. Теперь так иногда

называют и небольшую кластерную конфигурацию, собранную в одной стойке, и классические конструкции, объединяющие географически распределенные кластеры и другие ресурсы. Мы будем использовать термин «грид» только в этом последнем, «классическом» значении.

Три критерия того, что распределенная система является гридом, приведены в [6]: грид - это такая система, которая

- координирует использование ресурсов *при отсутствии централизованного управления* этими ресурсами;
- использует стандартные, открытые, универсальные протоколы и интерфейсы;
- обеспечивает высококачественное обслуживание, с точки зрения таких, в частности, характеристик как время отклика, пропускная способность, доступность и надежность.

Пользуясь определением грида, приведенным в самом начале статьи, и этими критериями можно сразу сказать, что такие известные системы управления кластерами, как Sun Grid Engine или Portable Batch System (PBS) нельзя рассматривать как полномасштабный грид поскольку, среди прочего, они полностью контролируют отдельные компоненты системы (более подробное обсуждение см. в [6]).

Строго говоря, нельзя отнести к гриду и многие системы распределенных вычислений или удаленного запуска заданий даже, если они охватывают географически обширные области. Понятие «системы распределенных вычислений» более широкое и включает, в частности, более простые, чем грид системы запуска заданий на удаленных компьютерах и сбора результатов. Чтобы не утомлять читателя детальным анализом, обратимся опять к аналогии с энергосетями. Помимо крупных объединенных национальных и международных энергосистем, с развитой инфраструктурой и строгой стандартизацией, существуют и другие производители электроэнергетики, например:

- солнечные электростанции (раскиданы по многим регионам мира, слабо связаны друг с другом, конструкция каждой очень индивидуальна, никаких общих стандартов);
- локальные электросети на основе использования энергии ветра, приливов, рек и т.п. (унификация параметров только в пределах этой небольшой сети, упрощенная система подключения пользователей, оплаты за потребление энергии).

Упрощенно говоря, и многие системы распределенных вычислений аналогичны таким специфическим и/или локальным энергосетям и потому не являются гридом в точном смысле этого термина. Необходимо подчеркнуть, что сам по себе факт несоответствия таких систем критериям принадлежности к гриду, не означает, что они хуже (или лучше) грид-систем – просто они отвечают другим потребностям и в разных ситуациях могут оказаться более или менее подходящими для решения конкретных задач. Хорошее обсуждение этого вопроса можно найти в статье [7]. Приведем несколько примеров таких систем:

SETI@home (Search for Extra-Terrestrial Intelligence At Home - поиск внеземного разума на дому) — научный некоммерческий проект [8] распределённых вычислений, использующий свободные ресурсы на компьютерах добровольцев для поиска радиосигналов внеземного разума. Каждый пользователь персонального компьютера, подключённого к сети Интернет, может помочь проекту. Для этого на своем ПК требуется установить ПО, которое во время простоя (точнее, когда работает экранная заставка – screen saver) предоставляет ресурсы компьютера для анализа данных радиотелескопа Аресибо с целью поиска сигналов, которые можно интерпретировать как искусственные.

При этом компьютер запрашивает через Интернет у центрального сервера некоторую порцию данных, обрабатывает их с помощью установленного ПО и отправляет результаты серверу.

Важной особенностью проектов типа SETI@home является нечувствительность к времени, за которое может быть получен результат.

X-Com - система организации распределенных вычислений [7], построенная на основе модели "ведущий/ведомые" и клиент-серверной архитектуры с двумя типами базовых компонентов:

- сервер X-Com - центральная часть системы, отвечающая за разделение исходного набора данных на блоки, распределение заданий между узлами, координацию работы подчиненных узлов, контроль целостности результата расчета, сбор результата в единое целое.
- узел X-Com - любая вычислительная единица (рабочая станция, узел кластера, виртуальная машина), на которой происходит расчет прикладной программы.

Отдельные блоки входных данных передаются от сервера на узлы, где и происходит расчет, а полученные на узлах результаты передаются обратно на сервер. Узлы отвечают за обработку блоков входных данных, запрос новых заданий для расчета от сервера, передачу результатов расчета на сервер. Взаимодействие между узлами и сервером в X-Com происходит по протоколу HTTP, что позволяет подключать к системе практически любые вычислительные мощности, имеющие доступ к сети. Отметим, что для использования системы необходимо иметь учетные записи на всех узлах X-Com (но не требуется прав привилегированных пользователей). Любая прикладная программа, работающая в рамках системы X-Com, разбивается на две части: серверную и вычислительную. Серверная часть прикладной программы управляет формированием наборов данных для расчета на узлах; вычислительная часть представляет собой основной расчетный модуль для обработки наборов данных.

В отличие от проектов типа SETI@home, задания, запущенные посредством X-Com, выполняются не во время простоя компьютерного ресурса, а в соответствии с приоритетами учетной записи на данном ресурсе, от имени которой работает программное обеспечение этого узла X-Com.

OurGrid/MyGrid - система распределенных вычислений [9], принципы построения которой сходны с предыдущей. Основным отличием этой системы от XCom является то, что ПО, соответствующее и серверу, и узлу XCom, устанавливается на каждый компьютер. Таким образом, все компьютеры, входящие в систему, оказываются равноправными – каждый может работать и как сервер и как рабочий узел. Такие системы называются одноранговым (peer-to-peer) гридом. Мы не будем здесь подробно обсуждать эту весьма важную тему; интересующиеся могут ознакомиться со статьей, посвященной одноранговым системам распределенных вычислений в сборнике [10], и с подборкой статей на сайте проекта [9]. Этот пример показывает условность и классификаций, и названий – хотя эта система вряд ли удовлетворяет выше приведенным критериям грида (сформулированным одним из основоположников грид-технологий), однако это слово присутствует уже в самом названии системы.

Конечно, приведенные примеры далеко не исчерпывают множество типов систем распределенных вычислений – даже краткое рассмотрение всех типов и проектов в этой области увел бы нас далеко за рамки данного обзора. Для предварительного знакомства с разными проектами и тенденциями развития распределенных вычислений можно использовать, например, сайты [3].

В заключение этого обсуждения – что является и что *не* является гридом, - хочется сделать

следующее замечание. Грид-технология справедливо имеет репутацию одной из важнейших технологий будущего. Поэтому исследовательские группы с весьма неопределенным и слабым отношением к собственно грид-технологиям имеют тенденцию добавлять это слово в названия своих проектов. Естественно, такая ситуация делает более трудной задачу точного определения и разграничения различных технологий и подходов. Сложность ситуации усугубляется и тем, что коммерческие грид-решения зачастую называются совсем по-разному: в IBM и Computer Associates — вычислениями по требованию (On-demand Computing), в Hewlett-Packard — адаптивным предприятием (Adaptive Enterprise), в Sun Microsystems — стратегией N1, в Microsoft — Dynamic Systems Initiative. При всем различии этих названий цель решений примерно одинакова - создание условий для динамического распределения вычислительных ресурсов и ресурсов хранения. Общность между перечисленными подходами заключается еще и в том, что соответствующие им решения так или иначе реализуют архитектуры, ориентированные на сервисы (Service-Oriented Architecture, SOA). Об этой архитектуре мы подробнее поговорим в разделе 1.4. Но сначала остановимся на некоторых общих принципах архитектуры грид-систем.

1.4.3 Общие принципы архитектуры грид-систем

В дальнейшем мы сосредоточимся на случае глобального грида (в соответствии с классификацией, приведенной в предыдущем разделе). Для такой системы важнейшим условием эффективной работы является обеспечение взаимодействия (интероперабельности) между различными платформами, языками и программными средами. В сетевой среде интероперабельность подразумевает работу по общим протоколам. Протоколы регламентируют взаимодействие элементов распределенной системы, а также структуру передаваемой информации.

С другой стороны, как уже отмечалось ранее, и как мы более подробно расскажем ниже, функциональной базовой компонентой грид-системы является сервис (служба). Поэтому при формулировке общих принципов построения грида важно определить как структуру протоколов, на которых основана его работа, так и его архитектуру в терминах сервисов. Образно говоря, архитектура грид-систем имеет две «проекции» - протокольную и сервисную.

1.4.3.1 Структура (стек) протоколов глобального грида

Общая структура глобального грида описывается в виде стека (набора уровней или слоев) протоколов [1]. В такой модели каждый уровень предназначен для решения узкого круга задач и используется для предоставления услуг для более высоких уровней. Верхние уровни ближе к пользователю и работают с наиболее абстрактными объектами, тогда как нижние уровни сильно зависят от физической реализации грид-ресурсов. Полезно иметь в виду, что эта структура аналогична сетевой модели OSI (Open Systems Interconnection Reference Model; модель взаимодействия открытых систем), - абстрактной модели для сетевых коммуникаций и разработки сетевых протоколов (см., например, [38]). В левой части рис. 2 показаны уровни стека грид-протоколов, а справа - четыре аналогичных им уровней модели OSI (всего в стеке OSI семь уровней). Итак, стек грид-протоколов включает:

1. аппаратный уровень (Fabric Layer) составляют протоколы, по которым соответствующие службы непосредственно работают с ресурсами;
2. связывающий уровень (Connectivity Layer) составляют протоколы, которые обеспечивают обмен данными между компонентами базового уровня и протоколы аутентификации;
3. ресурсный уровень (Resource Layer) – это ядро многоуровневой системы,

протоколы которого взаимодействуют с ресурсами, используя унифицированный интерфейс и не различая архитектурные особенности конкретного ресурса;

4. коллективный (Collective Layer) уровень отвечает за координацию использования имеющихся ресурсов;
5. прикладной уровень (Application Layer) описывает пользовательские приложения, работающие в среде виртуальной организации; приложения функционируют, используя протоколы, определенные на нижележащих уровнях.



Рис.2 Стеки протоколов грид-системы и сетевой модели.

1.4.3.1.1 Аппаратный уровень: управление локальными ресурсами

Аппаратный уровень обеспечивает доступ к распределенным ресурсам, необходимый протоколам более высокого уровня. Как уже неоднократно отмечалось, спектр возможных ресурсов весьма широк - это могут быть компьютеры, устройства массовой хранения данных, каталоги, сетевые ресурсы и т. п.

При этом ресурс может быть логической сущностью (например, распределенной файловой системой) или физической (например, кластером компьютеров). Реализация такого ресурса может включать внутренние протоколы (например, NFS (Network File System) или протокол управления кластером), однако подобные протоколы не включаются в стек грида. Компоненты аппаратного уровня реализуют локальные операции, специфические для каждого данного ресурса (логического или физического). Этот уровень по своим функциям аналогичен каналному уровню модели OSI и, по существу, представляет собой набор интерфейсов для управления локальными ресурсами.

1.4.3.1.2 Связывающий уровень: коммуникации и безопасность

Коммуникационные протоколы связывающего уровня (Connectivity) должны обеспечивать надежный транспорт и маршрутизацию сообщений, а также присвоение имен объектам сети, а протоколы аутентификации этого уровня, основываясь на коммуникационных, предоставляют криптографические механизмы для идентификации и проверки подлинности пользователей и ресурсов. Инфраструктура поддержки включает централизованную выдачу сертификатов, управление сертификатами и ключами и т. д.

1.4.3.1.3 Ресурсный уровень: совместное использование ресурсов

Ресурсный уровень с помощью коммуникационных и аутентификационных протоколов, входящих в нижележащий связывающий уровень, проводит согласование методов безопасности, инициализацию и мониторинг ресурсов, и управление ими. Для доступа к

локальным ресурсам и дальнейшего управления ресурсный уровень вызывает соответствующие функции аппаратного уровня. Заметим, что протоколы ресурсного уровня предназначены исключительно для работы с локальными ресурсами, они не учитывают глобальное состояние системы. Этим занимается коллективный уровень, располагающийся выше.

Ресурсный уровень включает два основных класса протоколов:

- информационные протоколы, предназначенные для получения информации о структуре и состоянии ресурса, его конфигурации, текущей загрузке и политике (то есть, условиях) предоставления ресурсов (например, стоимости их использования);
- протоколы управления, обеспечивающие согласованность доступа к разделяемому ресурсу и определяющие необходимые операции, которые ресурс должен выполнить (скажем, инициализация процесса или доступ к данным).

1.4.3.1.4 Коллективный уровень: координация ресурсов

Протоколы коллективного уровня отвечают за взаимодействие всех элементов пула ресурсов, что и отражено в самом названии. В качестве примера глобальных функций и сервисов, реализуемых протоколами этого уровня, можно назвать службу каталогов, распределение ресурсов, планирование и брокерские услуги, службы мониторинга, диагностики, репликации данных, коллективной авторизации.

1.4.3.1.5 Прикладной уровень: запуск приложений в грид-среде

Этот высший уровень грид-архитектуры включает пользовательские приложения, которые исполняются в среде объединенных ресурсов. В процессе исполнения приложения используют протоколы нижележащих уровней, обеспечивающие доступ к необходимым службам, а также прикладные программные интерфейсы (Application Programming Interface – API), соответствующие данным протоколам.

Приложения могут вызываться через достаточно сложные оболочки и библиотеки. Эти оболочки сами могут определять протоколы, сервисы и прикладные программные интерфейсы, однако подобные надстройки не относятся к фундаментальным протоколам и сервисам, определяющим архитектуру грид-систем.

1.4.3.2 Архитектура сервисов распределенных систем и технологии ее реализации

На начальном этапе своего развития распределенные приложения взаимодействовали с помощью частных, специально разработанных (проприетарных) протоколов, а системные администраторы использовали оригинальные методы, чтобы ими управлять. Как уже отмечалось, за годы становления и развития этой технологии были выработаны - с различной степенью успеха - многочисленные стандарты, чтобы ослабить затраты на развертывание и обслуживание таких систем. Сегодня наиболее предпочтительными подходами к построению распределенных систем считаются сервисно-ориентированная архитектура (Service Oriented Architecture, SOA), технология веб-сервисов, и грид-стандарты (в первую очередь - Открытая архитектура грид-сервисов (Open Grid Services Architecture, OGSA)). Однако, свои преимущества имеет и объектно-ориентированный подход. Ниже мы даем краткое общее описание этих подходов и соотношений между ними. При написании этого раздела существенно использовалась статья [28].

1.4.3.2.1 Сервисно-ориентированная архитектура (SOA) распределенных систем

При рассмотрении термина «сервисно-ориентированная архитектура», полезно предварительно определить ключевые термины:

- **Архитектура** это формальное описание системы, определяющее ее цели, функции, внешне видимые свойства, и интерфейсы. Она также включает описание внутренних компонентов системы и их отношений, наряду с принципами, управляющими ее дизайном, функционированием и возможной последующей эволюцией.
- **Сервис (служба)** - программный компонент, к которому можно удаленно обратиться посредством компьютерной сети, и предоставляющая некоторые функциональные возможности запрашивающей стороне.
- **Сервисно-ориентированная архитектура (service-oriented architecture, SOA)** является основой построения надежных распределенных систем, которые в качестве услуг предоставляют функциональные возможности, с дополнительным акцентом на слабые связи между взаимодействующими сервисами.

Таким образом, SOA это архитектурный стиль, который подчеркивает реализацию компонентов системы как модульных сервисов, которые могут быть найдены и использованы клиентами. Вообще говоря, сервисы имеют следующие характеристики:

- Сервисы могут быть полезными каждый сам по себе, или они могут быть объединены, чтобы предоставить единый высокоуровневый сервис. Среди других достоинств, это позволяет многократно использовать уже существующие функциональные возможности.
- Сервис общаются со своими клиентами, обмениваясь сообщениями: фактически, сервисы и различаются/идентифицируются в соответствии с совокупностью сообщений, которые они могут принять и ответов, которые они могут направить запрашивающей стороне.
- Сервис может участвовать в таких процессах обработки запросов, при которых порядок, в котором сообщения посылаются и принимаются, влияет на результат операций, выполненных сервисом. Это называется "сервисной хореографией" (service choreography) [5].
- Сервис может быть полностью независимым, или может зависеть от существования других сервисов, или каких-либо ресурсов, например, баз данных. Простейшим примером первого типа является сервис, осуществляющий возведение в квадрат числа, посланного в запросе; примером второго типа является сервис, осуществляющий конвертацию валют – для этого ему необходим текущий валютный курс.
- Сервисы предоставляют информацию о своих возможностях, интерфейсах, политике и поддерживаемых протоколах связи. Детали реализации, например, язык программирования и платформа, на которой он реализован, не нужны клиентам для направления запросов и не предоставляются.

Рис. 3 иллюстрирует простой цикл взаимодействия сервисов, который начинается с того, что данный сервис оповещает о своем существовании и свойствах посредством известного сервиса регистрации (1), свойства и способы взаимодействия с которыми должны быть заранее известны клиентам.

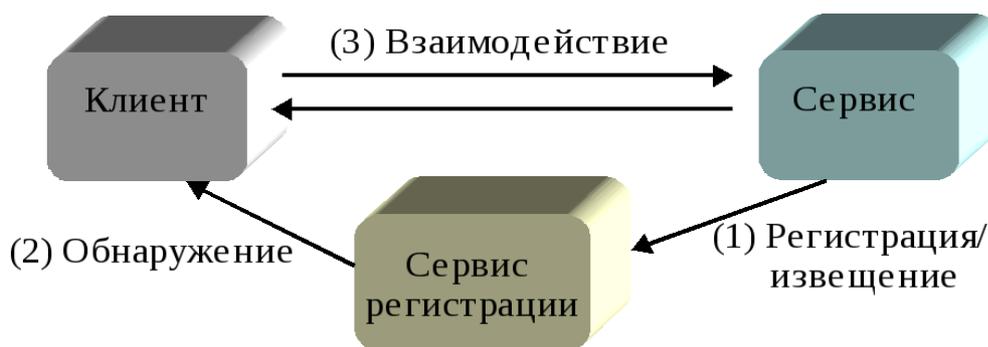


Рис.3 Взаимодействие сервисов в SOA-среде

Потенциальный клиент, который может быть другим сервисом (или человеком), делает запрос в сервис регистрации (2), чтобы найти сервис, который удовлетворяет его потребностям. Регистрационный сервис возвращает (возможно пустой) список подходящих сервисов; клиент выбирает один из них и передает ему запрос, используя любой взаимно распознаваемый протокол (3). В этом примере, сервис отвечает (4), передавая или результат требуемой операции или сообщение об ошибке. На рисунке показан самый простой случай. В реальной установке процесс может быть значительно более сложным. Например, данный сервис может поддерживать только HTTPS протокол, обслуживать только зарегистрированных пользователей, предлагать различные уровни сервиса различным пользователям, или требовать оплаты за использование. Представленный иллюстративный пример соответствует простому синхронному, двунаправленному способу обмена сообщениями, в то время как в реальной жизни взаимодействие может быть односторонним, или ответ может быть получен не от того сервиса, которому клиент посылал запрос, но от некоторого другого, которому он был передан для завершения обработки.

1.4.3.2.2 Принцип слабой связи

В определение SOA входит понятие слабой связи сервисов. Этот термин подразумевает, что взаимодействующие программные компоненты имеют минимальное знание друг о друге: они находят информацию, которая им нужна для взаимодействия непосредственно перед взаимодействием. Например, узнав о существовании сервиса, клиент может выяснить его возможности, условия предоставления услуг, его местоположение, его интерфейсы и поддерживаемые протоколы. Как только эти сведения получены, клиент может сразу же обратиться к сервису, используя любой взаимоприемлемый протокол. Достоинствами слабой связи являются:

- *Гибкость*: сервис может быть расположен на любом сервере, а при необходимости - перемещен. Пока ссылка на этот сервис есть в службе регистрации, предполагаемые клиенты будут в состоянии найти и использовать его.
- *Масштабируемость*: функциональные возможности сервиса могут быть расширены или сужены, поскольку при этом описание сервиса динамически меняется и, соответственно, изменяются и запросы.
- *Возможность модификации реализации*: при условии, что оригинальные интерфейсы сохраняются, реализация сервиса может быть обновлена без сбоев в обслуживании клиентов.
- *Отказоустойчивость*: Если возникают проблемы в работе сервера, программного компонента или сегмента сети, или сервис становится недоступным по любой другой причине, клиенты могут сделать запрос к службе регистрации для обнаружения другого сервиса, который предоставляет требуемые услуги.

Ясно, все эти преимущества имеют большое значение в динамической распределенной

среде. Однако, необходимо отметить, что при конкретных реализациях сервисов далеко не всегда удается достигнуть полной независимости клиентов от свойств сервисов. В реальности термин «слабосвязанный» является относительным и степень зависимости клиента и сервиса, при которой их можно назвать сильносвязанными, в значительной степени является субъективным решением.

1.4.3.2.3 Понятие состояния сервисов и сервисы без состояний

Одними из ключевых понятий в теории слабосвязанных сервисов являются понятия «состояния сервиса» и «сервис без состояний». Дело в том, что преимущества слабой связи вытекают из факта, что клиент может использовать любой сервис, который способен выполнить его запрос. Если выбор клиента ограничен единственным сервисом, тогда преимущества слабой связи существенно уменьшаются. В простом случае калькулятора или сервиса отслеживания курса акций является очевидным, что, как только клиент запросил и получил информацию, транзакция закончена, и у клиента нет никакой особой нужды при последующих запросах на аналогичные услуги обращаться к тому же самому сервису. С этой точки зрения, клиент и сервис связаны слабо.

Однако, для более сложных запросов, которые требуют нескольких шагов, дизайн сервиса мог бы быть таким, что сервис сохраняет в своей локальной памяти некоторую информацию ("состояние") о первом шаге, предполагая использовать ее, когда клиент входит с ним в контакт на следующем шаге. В этом случае, сервис обладает состоянием (stateful service), и клиент должен возвратиться к тому же самому сервису при следующем шаге. Это может привести к задержкам, если много клиентов используют один и тот же сервис или даже к отказу в обслуживании, если узел, являющийся главным сервисом, аварийно прекращает работу между шагами.

Более правильный подход при разработке сервисов основан на «сервисах без состояний» (stateless). Это подразумевает что при многошаговой обработке запросов:

В конце каждого промежуточного шага сервис должно вернуть клиенту достаточную информацию о состоянии, чтобы дать возможность любому сервису с соответствующими свойствами идентифицировать и продолжить обслуживание.

Клиент должен передавать информацию о состоянии любому сервису на следующем шаге обработки запроса.

Выбранный сервис должен быть в состоянии принимать и обрабатывать информацию о состоянии, поставляемую клиентом, независимо от того обрабатывал ли он сам запрос на предыдущем шаге, или это делал другой сервис.

Рис. 4 показывает клиента, делающего запрос, для обработки которого требуется три шага и несколько сервисов, каждый из которого может быть способен к обработке любой части или всего запроса.

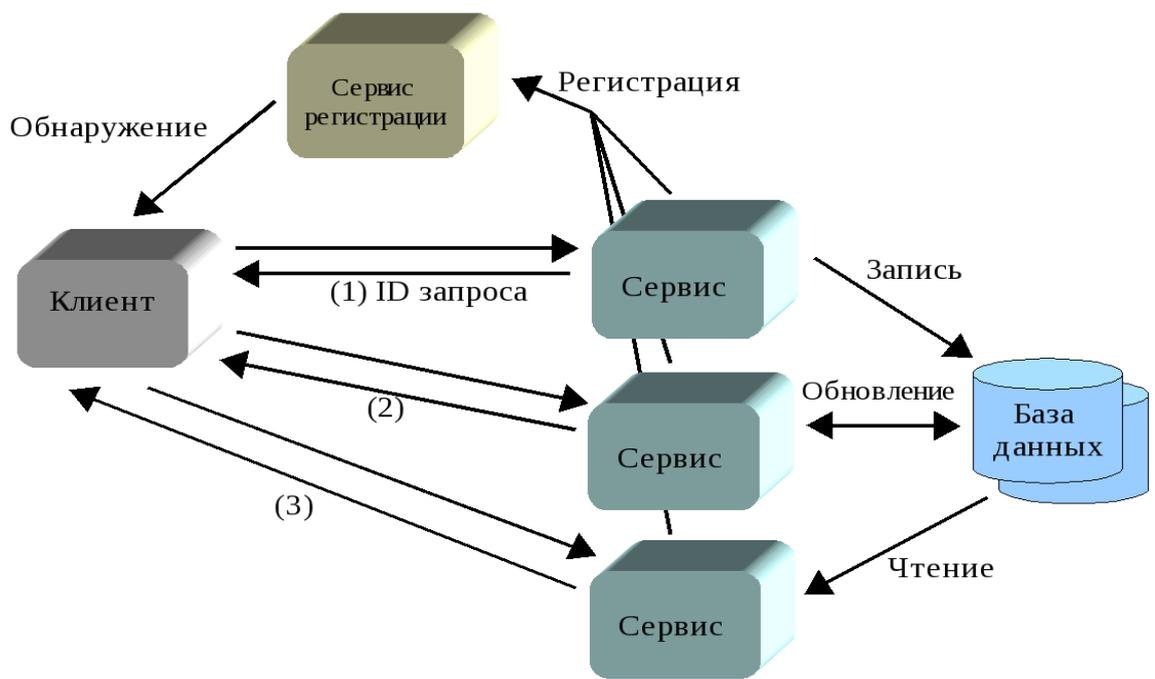


Рис. 4 Мношаговое взаимодействие клиента и сервисов

Сервис, который обрабатывает шаг 1, сохраняет детали обработки текущего запроса в базе данных, и возвращает запрашиваемую информацию клиенту, наряду с идентификатором запроса. Клиент может запросить подтверждение со стороны пользователя перед передачей этого идентификатора другому сервису, который использует его, чтобы отыскать информацию о состоянии в базе данных и инициализирует шаг 2. Этот сервис затем обновляет базу данных и возвращает дополнительную информацию клиенту. Наконец, клиент передает операционный идентификатор третьему сервису вместе с запросом на окончание всей обработки.

Большинство нетривиальных приложений требует доступа к некоторому количеству информации о состоянии, и вопрос не о том, должны ли состояния существовать, а о том, где они должны храниться. В подходе, описанном выше, состояние обработки запроса отделено от сервиса, который эту обработку осуществляет – обеспечивая, тем самым, слабую связь сервисов и клиента. Для того чтобы максимально уменьшить объем информации о состояниях, которой обмениваются клиент и сервисы, существенные детали об обработке запроса сохраняются в базе данных. При все участвующие сервисы должны быть в состоянии обратиться к базе данных и получить требуемую информации на основе идентификационного номера клиента/запроса, который можно легко передать от клиента сервисам.

1.4.3.2.4 Веб-сервисы

Большинство людей знакомо с доступом во всемирную паутину через веб-браузеры, которые обеспечивают ориентируемый на пользователя интерфейс к информации (веб-страницам) и к ориентированным на человека сервисам типа сетевых аукционов и розничных продаж (интернет-магазинам). Когда пользователь запрашивает веб-страницу, запрос обрабатывается удаленным веб-сервером, который возвращает информацию на языке разметки гипертекста (HTML). Это позволяет браузеру представить ее используя соответствующие шрифты, цвета и изображения, все то, что делает восприятие человеком этой информации более простым и удобным.

Веб-сервисы (или веб-службы) – это распределенные программные компоненты, идентифицируемые своим сетевым адресом, интерфейс которых описан на специальном

«диалекте» языка XML (eXtensible Markup Language), а именно WSDL (Web Service Description Language). Другие программные системы могут взаимодействовать с веб-сервисами согласно этому описанию посредством сообщений, основанных на другом «диалекте» XML - SOAP, и передаваемых с помощью интернет-протоколов (обзор технологий веб-сервисов можно найти, например, в [5]). В интернет-магазинах, например, веб-сервисы, которые могут быть реализованы на географически далеко удаленных друг от друга серверах, могут обеспечивать ведение счетов, контроль наличия товаров на складе, составление «корзины» покупателя и службу авторизации кредитной карты. Каждая из этих функций может быть многократно вызвана в ходе одной покупки. Веб-сервисы предоставляют детальную информацию об их функциях и о интерфейсах, но не о деталях реализации и платформе; таким образом клиент и сервис, которые поддерживают общие протоколы коммуникации, могут взаимодействовать независимо от платформ, на которых они выполняются или языков программирования, на которых они написаны. Это делает веб-сервисы особенно подходящими для распределенной гетерогенной среды.

Ключевыми спецификациями, используемыми веб-сервисами являются:

- XML - язык разметки для того, чтобы обмениваться структурированными данными;
- SOAP – основанный на XML протокол для спецификации содержания сообщений, служебной «обертывающей» информации («конверт сообщения») и информации, необходимой для правильной обработки сообщений;
- WSDL (Web Services Description Language - язык описания веб-сервис) – основанный на XML, используемый для описания атрибутов, интерфейсов и других свойств веб-сервис. Документ WSDL используется потенциальным клиентом, чтобы узнать о сервисе и способах обращения к нему.

Хотя веб-сервисы, вообще говоря, могут использовать разные транспортные протоколы, и даже могут предлагать клиентам выбор, самым распространенным является SOAP поверх протоколов HTTP или HTTPS. Это важно поскольку использование HTTP и HTTPS обычно не вызывает проблем с обходом систем сетевой защиты в организациях, которые разрешают двунаправленный трафик HTTP.

1.4.3.2.5 Веб-сервисы и SOA

SOA и веб-сервисы являются «ортогональными» понятиями: сервисная ориентация – это архитектурный стиль, а веб-сервисы - технология выполнения. Они, конечно, могут использоваться совместно – как это часто и случается, но они взаимно независимы.

Например, хотя принято считать, что SOA предназначена для распределенных систем, она может использоваться и для одиночных компьютеров, когда сервис соответствует индивидуальным процессам с четкими интерфейсами, которые взаимодействуют используя внутренние каналы связи, или для кластера персональных компьютеров, где они могли бы общаться посредством высокоскоростной локальной сети.

Аналогично, веб-сервисы хорошо подходят в качестве строительных блоков SOA-среды, но в их определении нет ничего, что обязательно требует воплощение принципов SOA. Хотя отсутствие состояний зачастую рассматривается как ключевая характеристика веб-сервис, нет никаких технических причин, по которым они обязательно должны быть без состояний – это определяется выбором разработчика и архитектурным стилем среды, в которой сервис должен работать.

1.4.3.2.6 Веб-сервисы и виртуализация

Легкость, с которой веб-сервисы могут быть реализованы и возможность обращаться к ним локально или удаленно и независимо от платформы на которой работает клиент, привела к тому, что они были широко приняты администраторами систем как «агенты виртуализации», которые обеспечивают общие интерфейсы управления к различным ресурсам. Например, веб-сервиса может быть спроектирована так, чтобы "представлять"

специфическое устройство или обычное приложение, принимая запросы управления и контроля через стандартизированные интерфейсы, с ресурсом общаясь посредством его «родного» интерфейса, и возвращая результат запрашивающей стороне опять в стандартном формате. Рис. 5 иллюстрирует, каким образом единая консоль может управлять совокупностью разнообразных ресурсов посредством уровня виртуализации на основе веб-сервис, каждая из которых предоставляет консоли стандартизированные интерфейсы для взаимодействия со связанным с ней ресурсом.

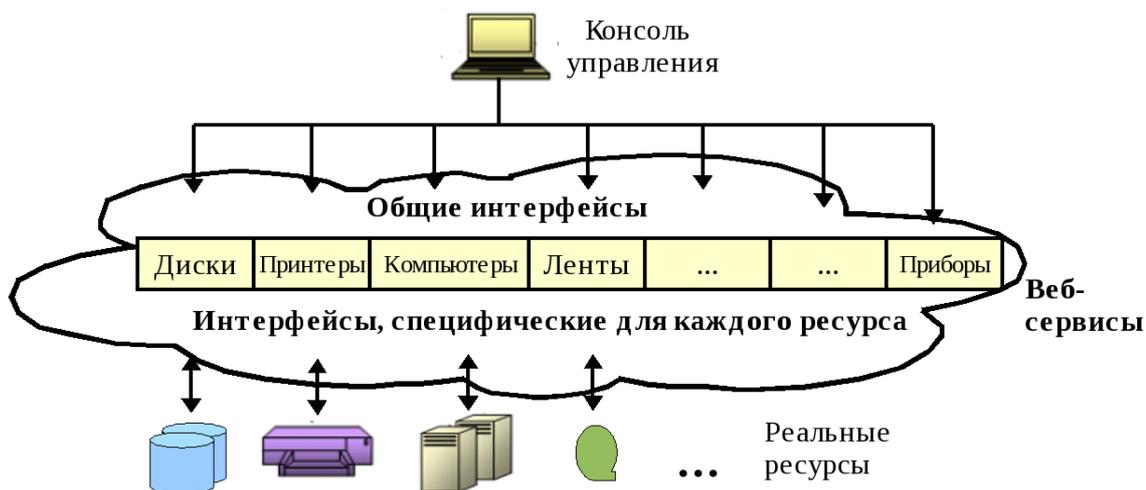


Рис.5 Виртуализация ресурсов с помощью Веб-сервисов

1.4.3.2.7 Сервисно-ориентированный грид

Работа грид-систем опирается на программное обеспечение промежуточного уровня: программные компоненты и протоколы, которые обеспечивают требуемый контролируемый доступ к ресурсам. На первом этапе своего существования грид-системы строились или на основе специально разработанных общедоступных компонент или на основе закрытых (проприетарных) технологий. Хотя различные общественные и коммерческие решения были успешны в своих областях применения грида, каждое со своими сильными и слабыми сторонами, они имели ограниченный потенциал как основы для грида нового поколения, который должен быть масштабируемым и интероперабельным, чтобы удовлетворять потребности широкомасштабных научных и производственных проектов.

В последние годы стало ясно, что есть значительное перекрытие между целями вычислительного грида и преимуществ сред, основанных на SOA и веб-сервисах. Быстрый прогресс в технологии веб-сервисов и разработке соответствующих стандартов обеспечили эволюционный путь от жесткой и узко-направленной архитектуры грид-систем первого поколения к стандартизированным, сервис-ориентированным гридам, гарантирующим стабильно-высокое качество обслуживания пользователей (грид промышленного уровня).

На рис. 6 показана схематично представлен простой сервисно-ориентированный грид, в котором сервисы используются и для виртуализации ресурсов, и для обеспечения других функциональных возможностей грида.

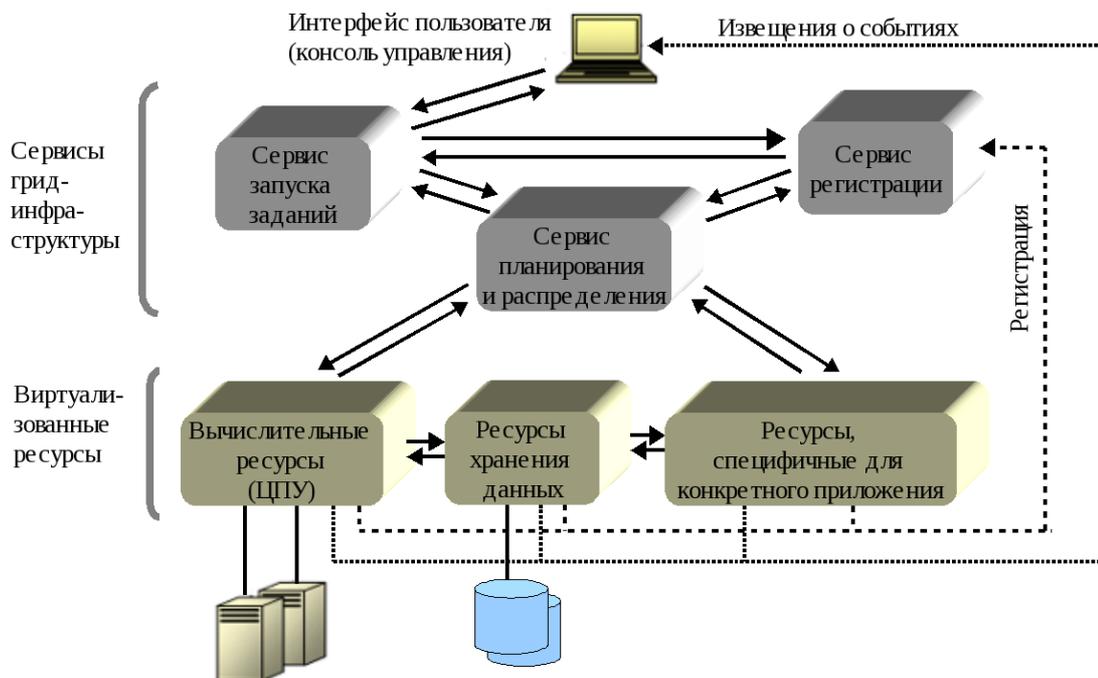


Рис. 6 Упрощенная схема сервисно-ориентированного грида

На схеме показана единая консоль и для запуска заданий в грид-среду, и для управления грид-ресурсами. Программное обеспечение интерфейса пользователя (консоли) обращается к сервису регистрации, чтобы получить информацию о существующих грид-ресурсах. Затем пользователь посредством консоли входит в контакт с сервисами, «представляющими» (виртуализующими) каждый ресурс, чтобы запросить периодическое получение данных о работе ресурсов и получение извещений о существенных изменениях в их состоянии (например, если ресурс становится недоступным или сильно загруженным).

Пользователь направляет запрос на запуск задания в службу запуска, которая передает запрос службе распределения заданий (часто называемой планировщиком). Служба распределения контактирует со службой, представляющей приложение, и запрашивает информацию о требованиях к ресурсам для выполнения задания. Затем служба распределения запрашивает у службы регистрации информацию о всех подходящих ресурсах в гриде и напрямую контактирует с ними, чтобы убедиться в их доступности. Если подходящие ресурсы доступны, планировщик выбирает наилучшую доступную совокупность ресурсов и передает информацию о их сервису приложения с запросом на начало выполнения. В противном случае планировщик ставит задание в очередь и выполняет его, когда необходимые ресурсы становятся доступными. Когда выполнение задания заканчивается, сервис приложения сообщает о результате планировщику, который извещает об этом сервис запуска заданий. Сервис запуска заданий, в свою очередь, уведомляет пользователя.

Отметим, что этот пример для ясности сильно упрощен: функционирование реального грида промышленного уровня является намного более сложным чем показано на схеме. Главным результатом его работы должна быть высокая степень автоматизации и оптимизации использования ресурсов в рамках грид-среды.

1.4.3.2.8 Open Grid Services Architecture (OGSA)

Конвергенция SOA и вычислительных грид-систем воплощена Глобальным грид-форумом (Global Grid Forum, GGF) в Открытой архитектуре грид-сервисов (Open Grid services

Architecture, OGSA). Основной документ [34], фиксирующий архитектуру OGSA, описывает общие принципы построения сервисно-ориентированного грида в терминах необходимых функциональных возможностей, например, выполнение заданий, управление ресурсами и данными, обеспечение безопасности. Основной целью этого и других документов, посвященных OGSA (см. [23]), является стандартизация интерфейсов и поведения основного (базового) набора грид-сервисов, чтобы они могли взаимодействовать как друг с другом, так и с грид-приложениями *независимо* от конкретных реализаций таких грид-сервисов. OGSA, в частности,

- определяет стандартные механизмы для создания, именованя и обнаружения постоянных и временных экземпляров сервисов;
- обеспечивает прозрачность местонахождения и взаимодействия по различным протоколам для экземпляров сервисов;
- описывает общую модель ресурсов, которая представляет собой абстрактное представление как физических ресурсов (таких как процессоры, процессы, диски и файловые системы), так и логических (экземпляров сервисов).

Для понимания терминологии необходимо отметить, что в рамках OGSA различают общее понятие *сервиса* и понятие *экземпляра сервиса*. Сервис - это абстрактное понятие, он определяется своим интерфейсом и способом взаимодействия с другими сервисами. При этом не имеется в виду, что сервис обязательно как-то реализован и реально функционирует. Напротив, реально функционирующий сервис является экземпляром того или иного сервиса.

При разработке OGSA учитывался ряд требований, полученных в результате анализа различных сценариев использования грид-систем, в том числе:

- интероперабельность (виртуализация ресурсов, общие способы управления и обнаружение ресурсов, стандартизация протоколов);
- разделяемый доступ к ресурсам;
- оптимизация выделения ресурсов;
- возможность запуска заданий на удаленных вычислителях;
- возможность манипуляции данными;
- снижение стоимости администрирования больших гетерогенных систем;
- безопасность (аутентификация и авторизация, возможность интеграции с различными системами безопасности, делегирование прав)
- надежность;
- гарантии качества обслуживания;
- простота использования и расширяемость (возможность расширения спектра применения);
- масштабируемость (отсутствие узких мест, препятствующих наращиванию объема ресурсов).

Кратко упомянем некоторые положения Открытой архитектуры грид-сервисов.

Грид-сервисы в OGSA. OGSA определяет грид-сервис как веб-сервис, который предоставляет набор корректно определенных интерфейсов, определенных на языке WSDL, и следует специфическим конвенциям для их создания и композиции сложных распределенных систем. Интерфейсы определяют способы обнаружения, динамического

создания службы, управления жизненным циклом, уведомления, управляемости. Конвенции определяют способ именования и возможность модернизации грид-служб. OGSA также определяет механизмы обновления «знаний» клиента о службе, в том числе, о поддерживаемых ею операциях и сетевых протоколах, которые клиент может использовать для связи с этой службой.

В отличие от традиционных веб-сервисов, которые позволяют обнаруживать и инициировать только постоянные (persistent) службы, OGSA также предусматривает поддержку временных (transient) экземпляров сервисов, инициируемых и завершаемых динамически (например, временный экземпляр службы может быть создан для обращения к хранилищу данных, для резервирования сетевых или процессорных ресурсов).

Таким образом, грид-сервис — это (возможно временная) служба на базе грид-протоколов, описанная посредством WSDL. Каждый интерфейс грид-службы определяет набор операций, которые инициируются путем передачи соответствующей последовательности сообщений.

OGSA описывает как создается грид-служба, как называется, как определено ее время жизни и т.д. OGSA предписывает базовое поведение, но не определяет конфигурацию службы и ее развертывание в грид-инфраструктуре. Другими словами, OGSA не касается вопросов реализации, парадигмы программирования, конкретных языков, механизмов и инструментальных средств реализации или операционной среды. Это позволяет модели грид-служб абстрагироваться от особенностей программного и аппаратного обеспечения, предлагая лишь набор стандартных интерфейсов высокого уровня. В частности, согласно OGSA, любой грид-сервис должен реализовывать тип GridService, служащий основным определением интерфейсов в OGSA. Он аналогичен базовому классу Object в объектно-ориентированных языках в том, что он инкапсулирует базовое поведение компонентной модели. В качестве базовых средств коммуникации GridService предусматривает передачу сообщений, ориентированных на документы, а также механизм RPC (remote procedure call — «удаленный вызов процедур»). При передаче сообщений входными и выходными объектами являются XML-документы, в то время как RPC использует более строго определенные API, но зато обеспечивает более высокую производительность.

Дескрипторы Grid-сервисов. Экземпляры служб в рамках OGSA могут создаваться с помощью операции CreateService или вручную, и уничтожаться динамически, когда истекает время, на которое она создавалась, или в любой момент с помощью операции Destroy. Операция CreateService возвращает дескриптор грид-службы (Grid Service Handle, GSH), который представляет собой глобально уникальный указатель (URL), определяющий имя экземпляра службы и отличающий его от других экземпляров. GSH не несет в себе всей информации, достаточной для того, чтобы клиент мог напрямую взаимодействовать с экземпляром службы. Поэтому GSH необходимо отобразить на ссылку на грид-службу (Grid Service Reference, GSR), которая содержит информацию, необходимую клиенту для связи со службой. Такая двухступенчатая структура позволяет GSH оставаться неизменной в течение всего срока существования экземпляра грид-сервиса, в то время как GSR может измениться (например, если служба «переносится» на другой компьютер). Если такое происходит, клиент должен отобразить GSH службы на новую ссылку.

Модель уведомлений. Эта часть OGSA описывает доставку сообщений о наборах данных о структуре службы и об изменениях ее состояния от источника адресату. Источник уведомлений — это экземпляр грид-службы, который реализует интерфейс (portType в терминологии языка WSDL) NotificationSource и посылает сообщение с уведомлением. Адресат — экземпляр грид-службы, который реализует интерфейс NotificationSink и получает сообщение с уведомлением. Чтобы инициировать уведомление от конкретной

грид-службы, пользователь вызывает операцию `subscribe` в соответствии с интерфейсом источника уведомления, передавая ему дескриптор GSN службы-получателя уведомления.

В любом случае, подход на основе OGSA предполагает использование стандартов, гарантирующих функциональную совместимость различных реализаций грид-систем. По мере того как спецификации OGSA будут публиковаться и становиться зрелыми (см. Дорожную карту OGSA [35]), ожидается, что программные средства промежуточного уровня различных поставщиков будут обладать OGSA-совместимыми функциональными возможностями и интерфейсами. В этом случае разработчики грид-систем смогут объединять компоненты различных поставщиков ПО для построения масштабируемых, функционально-совместимых (интероперабельных), надежных и легко управляемых грид-систем.

Важно отметить, что при определении рабочими группами GGF/OGF новых стандартов, OGSA будет использовать, как уже отмечалось в разделе 1.3.3, существующие и появляющиеся стандарты из всех сегментов веб-сервисного сообщества, чтобы обеспечить потребности различных категорий пользователей. Например, на первом этапе спецификации OGSA методы управления грид-системами определяются на основе веб-сервисного стандарта WSDM (Web Services Distributed Management) [40] международной организации OASIS [36], с возможностью использования также альтернативного набора спецификаций, основанного на стандарте ассоциации DMTF [37]. А вот стандарты WSRF (Web Services Resource Framework), которые определяют способы представления и управления ресурсами/сервисами с состоянием, и WS-Notification, который обеспечивает общий механизм извещений и подписки на уведомления о событиях, были с самого начала разработаны и утверждены в качестве стандарта OASIS для нужд построения грид-систем (более подробно об этих стандартах – в следующем разделе).

Краткий обзор OGSA можно найти в [42].

1.4.3.2.9 Грид-спецификации WSRF (Web Services Resource Framework) и WS-Notification

Цель, преследуемая при создании этих спецификаций, заключается в сближении OGSA с веб-сервисами и SOA. С помощью средств, соответствующих этим спецификациям, может быть реализован подход к моделированию и управлению состоянием в контексте веб-сервисов. Иными словами, делается попытка реализовать «состояние» — именно то, что отличает грид-сервисы от веб-сервисов. Подобный подход позволяет пользователям, находясь в контексте веб-сервисов, контролировать и изменять состояние доступных им ресурсов.

WSRF включает следующие спецификации

- WS-Resource Lifetime - определяют способы управления жизненным циклом ресурса и специфицируются Web-сервисы для ликвидации ресурса;
- WS-Resource Properties - определяют способы запрашивания и модификации ресурсов, описываемых XML-документами Resource Property;
- WS-ServiceGroup - определяют способы представления и управления коллекциями Web-сервисов и/или WS-ресурсами;
- WS-BaseFaults - определяется базовый XML-тип, используемый при обмене сообщениями в Web-сервисах для информирования о сбоях.

Спецификации WS-Notification не входят непосредственно в состав набора WSRF, но разработаны на его основе. Набор WS-Notification состоит из трех спецификаций:

- WS-BaseNotification - данная спецификация определяет интерфейсы производителя (NotificationProducers) и потребителя (NotificationConsumers) асинхронных

сообщений, а также основные выполняемые функции при рассылке сообщений и подписке на них, процессов приостановки/возобновления подписок и контроля срока подписки.

- WS-BrokeredNotification - данная спецификация позволяет объекту, не относящемуся к Web-сервису создавать асинхронные сообщения и рассылать их через особую посредническую службу (NotificationBroker).
- WS-Topics - организация и категоризация тем для подписки.

Этот список дает представление о наборе функциональностей, для которых определяются стандарты этими спецификациями. Поскольку это чисто технические документы, подробно мы их не описываем. Хорошее введение в «идеологию» WSRF можно найти в статьях [43].

1.4.3.2.10 Другие грид-стандарты

Заметим, что некоторые грид-проекты, не могут дожидаться завершения работ над WSRF и пользуются альтернативными спецификациями, например, Basic Profile (BP1.0) от WS-Interoperability, Web Services Grid Application Framework (WS-GAF, North-East Regional e-Science Centre, www.neresc.ac.uk/ws-gaf) и WS-I+ (Open Middleware Infrastructure Institute, www.omii.ac.uk).

WS-Eventing. Предложенный Microsoft, BEA, Tibco, Sun и CA стандарт WS-Eventing обеспечивает общий метод взаимодействия веб-сервисов. Объявленная цель создания WS-Eventing заключается в поддержке обмена данными о событиях по схеме подписки/публикации. Формальная модель WS-Eventing строится на основе схемы XML, спецификации WSDL и поддерживает SOAP. В нем описывается протокол, позволяющий веб-сервисам предоставлять подписку «на себя» и подписываться на другие сервисы. Как указано в разд. 1.3.2, сейчас наметился процесс сближения этого стандарта с WSRF/WS-N.

Стандарт безопасности. Добавим еще, что фактическим стандартом безопасности в грид является Grid Security Infrastructure (GSI). Подробнее о нем будет рассказано во части II. В некоторых проектах исследуются альтернативные решения, которые могут повлиять на стандарты GSI. Например, в рамках проекта GridShib (<http://grid.ncsa.uiuc.edu/GridShib>) создаются новые механизмы и стратегии распределенной аутентификации, позволяющие виртуальным организациям в грид интегрироваться с традиционной инфраструктурой корпоративной безопасности.

В заключение рассказа о стандартизирующих спецификациях, хочется подчеркнуть, что если кто-то создает свою грид-систему, совсем не обязательно следовать каким-либо спецификациям. Более того, вполне может оказаться, что созданная таким образом оригинальная система будет работать даже эффективнее созданной по спецификациям – всегда платой за универсальность (интероперабельность) является пониженная производительность. Однако, необходимо четко понимать, что при этом не удастся развивать такую систему за счет интеграции сторонних продуктов (из-за несовместимости) и если возникнет необходимость «выхода во внешний мир», то есть включения такой грид-системы в качестве сегмента во внешнюю грид-среду, - это может потребовать значительных усилий, а показатели эффективности взаимодействия «с внешним миром» могут оказаться весьма низкими.

Обсуждение достоинств и недостатков (в том числе, вопросов производительности) систем на основе веб-сервисных технологий мы продолжим в следующем разделе.

1.4.3.2.11 Сервисно-ориентированные и объектно-ориентированные системы: сходство и различия, преимущества и недостатки

Выбор типа связи между компонентами распределенной системы - жесткий или свободный, - которому отдается предпочтение, соответственно, в объектно-ориентированном и сервисно-ориентированном подходах, интенсивно обсуждается специалистами и практиками, в том числе на ряде интернет-форумов. Мы сосредоточимся на различие этих подходов с точки зрения крупномасштабных распределенных грид-систем.

Традиционно, в распределенных объектно-ориентированных системах типа CORBA [45] и DCOM [46] программные компоненты с определенными функциональными возможностями рассматривались как распределенные объекты. В этой парадигме, клиенты имеют статическое знание об объектах, экземпляры объектов инкапсулируют данные, и взаимодействие происходит посредством синхронного вызова удаленных методов с использованием протоколов специального назначения. Это подразумевает, что взаимодействующие объекты являются сильно связанными, с жестко регламентированными методами запросов и ответов. В этой парадигме, объект с инкапсулированными данными является центральной фигурой. Объекты имеют тенденцию быть «мелкозернистым» отображением чего-то реально существующего. Другими словами, если речь идет о какой-либо сравнительно сложной системе, объекты имеют тенденцию отображать не ее общие свойства, а каждую из ее небольших частей, подсистем.

С другой стороны, в сервисно-ориентированной среде сервисы слабо связаны: они взаимодействуют с помощью обмена сообщениями, которые могут быть односторонними, синхронными или асинхронными. Этот подход стимулирует такой дизайн систем, в которых запросы и ответы не обязательно обрабатываются одним и тем же набором взаимодействующих компонент. В качестве примера, рассмотрим клиента, который обращается к сервису А, чтобы инициализировать заказ. Сервис А может проверить наличие товара, обращаясь к сервису Б, а сервис Б возможно обратится к сервису С, который формирует заявку и отправляет ее клиенту для подтверждения. Это простой пример, который иллюстрирует понятие свободной связи: запрос может быть направлен одному сервису, а ответ получен от другого. В этой парадигме, сервис является центральной фигурой. Вообще говоря, сервисы имеют тенденцию быть более «крупнозернистым» (общим, грубым) отображением реальных сущностей, чем объекты.

При текущей тенденции строить распределенные системы только на основе веб-сервисов, в подходе, основанном на распределенных объектах, зачастую видят одни недостатки. Это, однако, не так: достоинствами объектно-ориентированного подхода является зрелость существующих реализаций, эффективное взаимодействие компонент, модель инкапсулирования состояний и поведения объектов и это хорошо подходит для некоторым приложениям. Поэтому объектно-ориентированные распределенные технологии получили распространение в промышленных приложениях – для распределенных систем, например, финансовых учреждений, организаций здравоохранения, телекоммуникационных предприятий.

С другой стороны, хотя ясно, что SOA и веб-сервисы будут главной линией развития в обозримом будущем, нужно понимать, что они все еще имеют множество серьезных проблем, которые еще надо преодолеть, включая:

- пока все еще незрелую технологию,
- разночтения и некоторый беспорядок в области стандартов,
- существенные «накладные расходы» при работе Веб-сервисов, связанные сериализацией/десериализацией (преобразованием программного кода в XML-текст и обратно) и анализом SOAP-сообщений и документов XML при каждом обмене сообщениями.

Особенно существенным является последний недостаток, поскольку он негативным образом сказывается на производительности распределенных систем. Как всегда, за универсальность надо платить падением эффективности! Однако платформи-

независимость является столь важным доводом в пользу SOA и Веб-сервисов, что большинство крупных ИТ-фирм усиленно продвигают на рынок эту архитектуру и технологию.

К сожалению, наиболее известные в рамках объектно-ориентированного подхода спецификации CORBA обеспечивая хорошую производительность построенных на их основе реализаций, также несвободны от недостатков. Главный из них – чрезмерная громоздкость и недостаточная четкость этих спецификаций. Поэтому ряд существовавших грид-проектов на основе CORBA не получили значительного развития.

Сравнительно недавно появилось новое поколение ППО на основе объектно-распределенных технологий – ICE (Internet Communications Engine) [47]. Оно сочетает (и даже превосходит) высокую производительность CORBA с сравнительной простотой спецификаций, удобным языком описания интерфейсов, поддержкой основных языков программирования, развитым набором средств для построения распределенных систем и с рядом других достоинств.

Важно отметить, что, в то время как каждый архитектурный стиль имеет присущие ему достоинства и недостатки (в значительной степени преимущество того или иного стиля зависит от конкретной задачи, для которой они используются), оба стиля обеспечивают такие важные свойства, как:

- модульность, что, в свою очередь, позволяет
 - наращивать функциональные возможности без коренной переделки всей системы;
 - возможность построения крупных систем из более мелких;
- масштабируемость (увеличение числа однотипных компонент системы, например, вычислительных ресурсов грид-системы);
- устойчивость в работе;
- способности скрыть детали выполнения запросов за четкими интерфейсами;

Эти свойства позволяют создавать хорошо организованные распределенные среды для выполнения прикладных заданий.

1.4.3.2.12 Разработка систем на основе SOA

Как мы видели, сервисная ориентация следует таким общим программным принципам архитектуры, как инкапсуляция, модульность и отделение интерфейсов от их реализации. Кроме этого, SOA вводит понятия сервисной хореографии, сервисных репозиторий и взаимодействий, основанных на сообщениях и реакции на события. Чтобы строить систему, которая соответствует принципам SOA, нужно разделить общий алгоритм выполнения какой-либо работы (бизнес-логику) на части, а рабочие элементы - на сервисы. Сервисы:

- определяются хорошо доступными, независимыми от реализации интерфейсами;
- слабо связаны и следуют концепции «прозрачности» по отношению к их местонахождению (то есть клиенту не нужно знать - где конкретно располагается сервис);
- инкапсулируют многократно используемые функциональные возможности.

Следующий список перечисляет шаги, которым необходимо следовать, чтобы создать сервисно-ориентируемую систему:

1. Идентифицировать различные части бизнес-логики для работы, которая должна быть выполнена и рабочие единицы (элементы, модули), которые могут выполнить каждую из частей бизнес-логики.
2. Описать функциональные возможности различных модулей работы в терминах сервисов. Реализовать сервис с состояниями или без состояний – в зависимости от требуемых функциональных возможностей.

3. Идентифицировать базовые сервисы инфраструктуры, которые необходимы для работы всех остальных сервисов (то есть, инфраструктуры в целом); решить - существуют ли готовые реализации таких сервисов или они должны быть созданы.
4. Идентифицировать одинаковые части функциональных возможностей различных сервисов (если таковые вообще имеются) и обеспечить, чтобы одинаковые функциональные возможности были реализованы как многократно используемые модульные сервисы; для этого использовать технологию подтипов и другие методики расширения.
5. Зафиксировать различные части функциональных возможностей, предоставляемых сервисами в терминах операций, осуществляемых на основе обмена сообщениями и дать сервисам осмысленные названия, которые должны отражать их функциональные возможности.
6. Определить представляющие интерес события, которые сервисы могут порождать или на которые они могут реагировать.
7. Разработать управляющие скрипты для осуществления сервисной хореографии.
8. Зарегистрировать сервисы в сервисе регистрации (или нескольких сервисах регистрации), чтобы участники рабочего процесса могли обнаруживать и взаимодействовать друг с другом.

Как обсуждалось ранее, шаги 1 - 4 являются общими для дизайна любой распределенной архитектуры, объектно-ориентированого или сервис-ориентированого. Шаги 5 и 6 осуществляются также в обоих подходах, хотя различным способом. Шаги 7 и 8 присущи только сервисно-ориентированному подходу.

1.5 Заключение к первой части

В этой части мы попытались определить – что такое грид, как он соотносится с родственными технологиями и какие типы грид-систем бывают. Еще раз отметим, что типы грид-систем различаются по

- степени охвата ресурсов (грид-системы масштаба крупного предприятия, регионального и глобального масштаба);
- общей структуре (одноранговые (peer-to-peer) и многогранговые грид-системы);
- назначению - характеру основных прикладных, которые решаются с помощью данного грида (вычислительный грид, грид для интенсивной обработки данных);
- типу архитектуры (SOA, OGSA, объектно-ориентированные системы);
- технологии/спецификации построения (проприетарные решения и протоколы, веб-сервисные, WSRF и т.д.).

Веб-сервисы и сервисно-ориентированный стиль архитектуры широко рассматриваются как основа для нового поколения распределенных приложений и инструментальных средств управления системами. В этой части мы ввели ключевые концепции, соотношение и преимущества этих двух технологий, и указали - как они могут быть совместно использованы для разработки высоко масштабируемых прикладных систем независимо от аппаратных средств и программных платформ. Мы также попытались показать, как тщательная разработка ключевых стандартов увеличивает привлекательность сервисно-ориентированных сред, и как они используются в качестве основы для построения интероперабельных грид-систем следующего поколения. Мы также указали на возможность использования объектно-ориентированного подхода при построении распределенных систем и обсудили его достоинства и недостатки в сравнении с сервисно-ориентированным подходом.

2 Основные функциональные подсистемы глобального грида

В этой части будут рассмотрены основные подсистемы промежуточного программного обеспечения (ППО), обеспечивающего полноценное и безопасное функционирование глобального грида. Более ограниченные системы (например гриды-системы масштаба предприятия или простые системы распределенных вычислений, упомянутые в разделе 1.4.2) могут иметь только часть представленных в этой части подсистем или использовать их упрощенный вариант с ограниченной функциональностью. Мы изложим – по возможности кратко – лишь принципы, концептуальные основы, на которых построены эти подсистемы, не входя в детали конкретных реализаций. Дальнейшую информацию, включая вопросы инсталляции, конфигурирования и пользовательских интерфейсов к рассматриваемым подсистемам можно получить в инструкциях к конкретному промежуточному программному обеспечению грид-систем. Однако следует отметить, что наиболее близко наше описание следует архитектуре ППО gLite [22], являющегося основой крупнейшей в настоящее время грид-инфраструктуры EGEE [13] (подробнее об этой инфраструктуре мы расскажем в Части 3). Поскольку мы не касаемся вопросов конкретных реализаций, мы специально не оговариваем, но подразумеваем, что такие реализации основаны на веб-сервисных технологиях, которые мы обсуждали в предыдущей части.

Как уже упоминалось в предыдущей части, грид-система включает в себя следующие основные структурные компоненты:

- совокупность компьютеров с установленными на них интерфейсами пользователя;
- совокупность Ресурсных центров, включающих в себя
 - вычислительные ресурсы;
 - ресурсы хранения данных;
- набор базовых грид-сервисов.

Обобщенная схема грид-инфраструктуры представлена на рис. 7.

Интерфейс пользователя (User Interface, UI) предназначен для обеспечения доступа пользователя к ресурсам грида (в общем случае произвольных распределенных систем иногда также употребляется термин "консоль управления", см. Часть 1). Через UI пользователь

- осуществляет запуск заданий на выполнение;
- пересылает данные с одного ресурса хранения данных на другой;
- контролирует процесс выполнения задания;
- получает результат выполнения задания.

Ресурсный центр может включать два типа ресурсов (или один из них):

- Вычислительный ресурс, на котором выполняется обработка данных; служба, представляющая вычислительный ресурс в гриде, называется Вычислительный элемент (Computing Element, CE).
- Ресурс хранения данных (Storage Element, SE), который обеспечивает хранение и транспортировку данных между аналогичными ресурсами и/или данным ресурсом и пользователем.

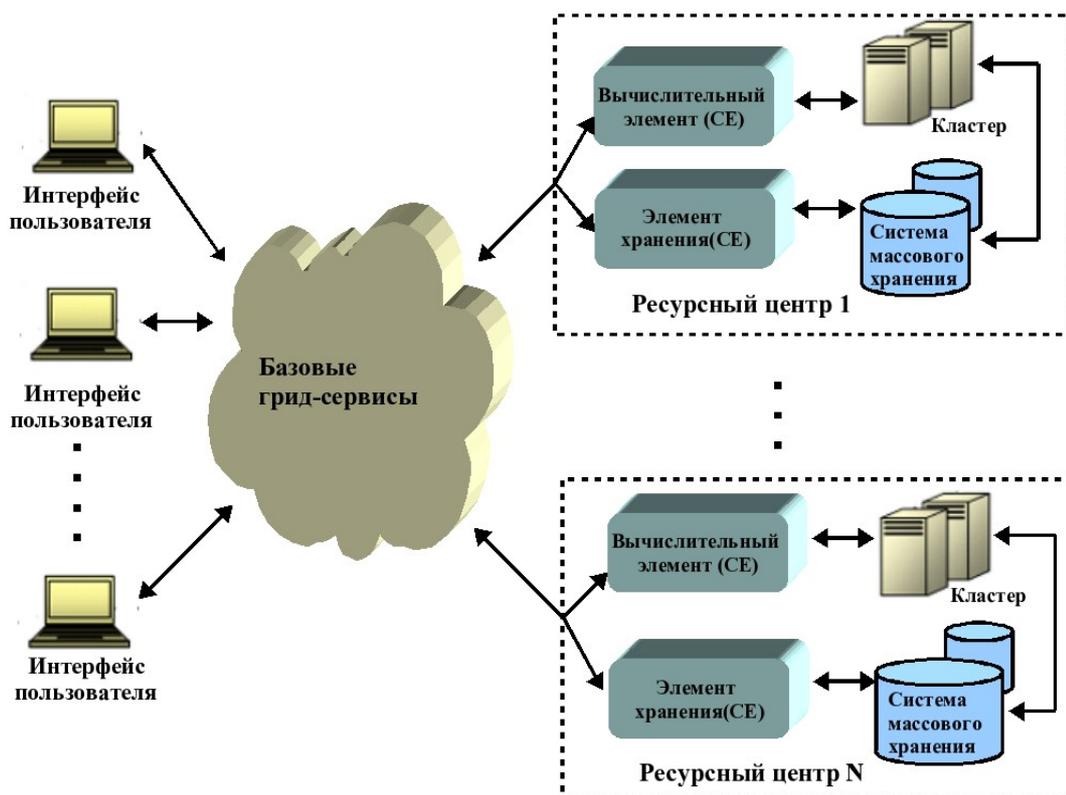


Рис.7 Обобщенная схема структуры грида

Базовые грид-службы обеспечивают работу всей грид-системы в целом; они подразделяются на следующие подсистемы:

- подсистема управления загрузкой (Workload Management System, WMS),
 - центральной является служба распределения заданий - брокер ресурсов (Resource Broker, RB);
- подсистема управления данными (Data Management System, DM)
 - к базовым службам этой подсистемы относятся службы каталогов:
 - служба файлового каталога,
 - служба каталога метаданных;
- подсистема информационного обслуживания и мониторинга грид-системы (Information System, IS)
 - служба регистрации и учета ресурсов грида,
- подсистема безопасности и контроля прав доступа (Grid Security Infrastructure, GSI)
 - служба выдачи и поддержки сертификатов (Certificate Authority, CA),
 - служба регистрации виртуальных организаций и пользователей,
 - служба управления виртуальными организациями и выдачи прокси-сертификатов (Virtual Organization Membership Service, VOMS),

- служба продления действия прокси-сертификата (MyProxy Service, MP);
- подсистема протоколирования (Logging and Bookkeeping, LB)
 - служба отслеживания статуса выполняемых заданий,
- подсистема учета (Accounting Subsystem, AS)
 - служба учета использования грид-ресурсов.

Благодаря ППО множество географически распределенных компьютерных ресурсов представляется пользователям единым ресурсом. Его роль в гриде может быть сравнена с ролью операционной системы в обычном персональном компьютере. Разумеется, для запуска прикладных задач в грид пользователь должен знать основы работы (команды, утилиты, форматы ввода/вывода) с этим ППО.

2.1 Краткое описание функционального назначения базовых подсистем

Базовые подсистемы грид-инфраструктуры выполняют следующие функции:

- *Подсистема управления загрузкой.* Задачей подсистемы управления загрузкой (WMS) является принятие запросов на запуск заданий, поиск соответствующих ресурсов и контроль их выполнения. Благодаря работе WMS, сложность управления приложениями и ресурсами в гриде скрыта от пользователей. Их взаимодействие с WMS ограничено описанием характеристик и требований запроса через ориентированный на пользователя язык спецификации высокого уровня - Язык описания заданий (Job Description Language, JDL), и к направлению такого запроса через предоставленные интерфейсы.
- *Подсистема управления данными.* Прикладное задание должно быть в состоянии обратиться к своим данным, независимо от фактического местоположения вычислительного ресурса. Необходимо обеспечить доступ к системам хранения разного типа, существующим в ресурсных центрах – от простых файловых систем до устройств массового хранения данных. Чтобы не иметь дело с особенностями каждого индивидуального устройства хранения данных, каждый включенный в грид-инфраструктуру ресурс хранения должен реализовывать стандартный интерфейс. В настоящее время, наиболее распространенным таким интерфейсом является Менеджер ресурсов хранения данных (Storage Resource Manager, SRM) [48]. Мы будем предполагать, что при управлении данными в грид-системе главным образом имеют дело с файлами, а не, например, с таблицами (последнее характерно для реляционных баз данных). Хранение данных в виде файлов является типичным в научных исследованиях и является удобным во многих других случаях, когда надо обрабатывать очень большие объемы данных (например, визуальную информацию).
- *Подсистема информационного обслуживания и мониторинга грид-системы* решает задачу сбора и управления данными о состоянии грида, получая информацию от множества распределенных источников – поставщиков. Подсистема предназначена для постоянного контроля функционирования грида и обеспечения своевременного реагирования на возникающие проблемы.
- *Подсистема безопасности и контроля прав доступа.* Чтобы грид был эффективной структурой для распределенных вычислений, пользователи, пользовательские процессы и службы грида должны работать в безопасной среде.

Для этого:

- взаимодействия между компонентами грида должны быть взаимно аутентифицированы (аутентификация - процесс подтверждения заявленных свойств объекта, на основании удостоверяющих его документов);
- любое действие должно совершаться только после соответствующей авторизации – сопоставления объекта, совершающего действие, и набора прав, предоставленных этому объекту для работы в грид-среде.
- *Подсистема протоколирования.* Подсистема протоколирования отслеживает процесс выполнения заданий, осуществляемый под управлением WMS. Она собирает извещения о событиях от различных компонентов WMS и обрабатывает их, чтобы представить обобщенное текущее состояние (статус) задания.
- *Подсистема учета.* Эта подсистема предназначена для учета использования вычислительных ресурсов (таких как процессорное время, использование оперативной памяти и так далее), и, в частности, может использоваться для формирования информации о стоимости использования данного грид-ресурса данным пользователем, если взаимоотношения пользователей и провайдеров ресурсов основаны на экономической модели.

2.2 Подсистема управления загрузкой

Взаимодействие с подсистемой управления (Workload Management System, WMS) загрузкой осуществляется с помощью интерфейса пользователя (UI), который позволяет делать запросы и управлять ими с помощью соответствующих программных модулей. Основными операциями, осуществляемыми с помощью UI, являются:

- формирование и получение списка ресурсов, подходящих для выполнения определенного задания;
- направление задания для выполнения на удаленном вычислительном ресурсе;
- проверка состояния направленного задания,
- отмена одного или более направленных заданий;
- получение выходных файлов завершеного задания с результатами его выполнения;
- получение учетной информации о направленных заданиях (продолжительность использования ресурсов, стоимость выполнения задания и так далее).

В некоторых грид-системах есть также следующие возможности:

- получение состояний задания в контрольных точках - для соответствующей категории заданий);
- запуск интерактивных заданий с помощью специального модуля-слушателя (listener), реагирующего на сообщения WMS о ходе выполнения;

2.2.1 Компоненты подсистемы управления загрузкой

После отправки с помощью интерфейса пользователя, запрос обрабатывается несколькими компонентами WMS. Их точный состав может варьироваться в зависимости от конкретного ППО, но обычно в него входят:

- менеджер загрузки (возможно с дополнительным прокси-сервером);
- планировщик/брокер ресурсов;

- адаптер заданий;
- модуль, ответственный за выполнение фактических операций по управлению заданиями (направление на выполнение, удаление задания, и т.д.);
- монитор log-файлов WMS.

Менеджер загрузки - основной компонент WMS. После получения запроса, он должен предпринять соответствующие действия, чтобы выполнить его. Чтобы сделать это, он взаимодействует с другими компонентами WMS, набор которых зависит от типа запроса. Для повышения производительности системы (количества запросов, которые могут быть обработаны за единицу времени), этот компонент дополняется **прокси-менеджером загрузки** ответственным за прием поступающих запросов от интерфейса пользователя (например, на запуск или удаление задания), анализ их корректности и, если результат анализа положителен, передачу запроса менеджеру загрузки.

Основным типом запросов к WMS являются запросы на выполнение заданий. А главными результатами их обработки подсистемой являются:

- правильный подбор грид-ресурса для выполнения конкретного задания на основании описания задания на языке JDL и информации о доступных ресурсах.
- направление задания на выбранный ресурс.

Первую задачу решает компонент, называемый **планировщиком** или **брокером ресурсов** (Resource Broker, RB). Существование подходящих ресурсов для данного задания зависит не только от состояния ресурсов, но также и от политики их использования, которой следуют администраторы ресурса и/или администратор виртуальной организации, к которой принадлежит пользователь.

Брокер ресурсов может следовать различной стратегии при распределении заданий по ресурсам:

- один крайний вариант стратегии - после поступления запроса на выполнение задания, как можно скорее подбираются наиболее подходящие ресурсы и, как только решение принято, задание передается на выбранный ресурс для выполнения («нетерпеливое планирование», в англоязычной литературе называется также push-model);
- другой крайний вариант - задания находятся в менеджере загрузки, пока какой-либо ресурс не становится доступным, после чего освободившемуся ресурсу подбирается наиболее соответствующее задание (из ожидающих в менеджере) и передается этому ресурсу для выполнения («ленивая политика планирования», pull-model);
- возможны промежуточные комбинированные варианты стратегии подбора ресурсов.

Для выполнения своих задач брокер использует базу данных о ресурсах, которая обновляется в результате либо получения уведомлений от ресурсов, либо активного опроса ресурсов подсистемой информационного обслуживания и мониторинга, или некоторой комбинации этих двух механизмов обновления. Кроме того, специальный компонент WMS, отвечающий за взаимодействие с информационной подсистемой, может быть сконфигурирован так, чтобы определенные уведомления об изменении состояний ресурсов могли вызвать начало работы брокера. Это обеспечивает реализацию «ленивой политики планирования» (например, после уведомления, что какой-то ресурс освобожден брокер инициализирует отправку на него ожидающего задания).

Другим важным компонентом, повышающим устойчивость работы WMS, является модуль управления **очередью заданий**, который дает возможность некоторое время хранить запрос на выполнение задания, даже если никакие ресурсы, соответствующие его требованиям, не доступны немедленно. Запросы, по которым не удалось сразу подобрать ресурсы, будут повторены либо периодически (при «нетерпеливом планировании») или как только уведомления о новых доступных ресурсах появляются в хранилище данных (при «ленивой политике планирования»). В противном случае, такие ситуации могли бы привести к немедленному прекращению выполнения задания из-за отсутствия подходящего ресурса.

Для того чтобы грид-система была способна обрабатывать не только независимые друг от друга задания, но и совокупности заданий, зависимости которых описываются направленным графом без замкнутых петель (см. раздел 1.2.3 и рис. 1), WMS должна иметь **мета-планировщик DAG**. Его основная цель состоит в том, чтобы управлять графом (то есть, DAG-запросом), определять, какие вершины графа являются свободными от зависимостей, и следить за выполнением соответствующего задания.

Перед тем, как отправить задание в ресурсный центр может потребоваться его дополнительная обработка - окончательная подготовка JDL-описания задания и создание обертывающего скрипта для него, который формирует соответствующую среду выполнения на рабочем узле соответствующего ресурсного центра (в частности, это необходимо для передачи файлов-контейнеров ввода/вывода; в программистской литературе такие файлы зачастую называются “sandboxes” – «песочницы»). Такая обработка выполняется отдельным модулем - **Адаптером заданий**.

Далее следует модуль, ответственный за выполнение фактических операций по управлению заданиями (направление на выполнение, удаление задания, и т.д.), выполняемых по запросам менеджера загрузки. В ППО gLite в качестве этого модуля используется **CondorC** [20].

Монитор log-файлов ответственен за просмотр сообщений модуля управления заданиями (например, в gLite - log-файлов CondorC), и перехват интересных событий относительно текущих заданий, то есть событий, касающихся изменений состояний задания (например, выполненное задание, отмененное задание, и т.д.), а также за инициализацию соответствующих действий.

2.2.2 Взаимодействие подсистемы управления загрузкой с другими грид-службами

Регистрационная и учетная информация относительно событий, сгенерированных различными компонентами WMS сохраняется в подсистеме протоколирования (раздел 2.6). Пользователь может узнавать о состояниях его задания, запрашивая эту подсистему (с помощью соответствующих команд – посредством интерфейса пользователя). Помимо активных запросов о состоянии задания, пользователь может также зарегистрироваться для того, чтобы получать уведомления об изменениях в состоянии заданий (например, об окончании выполнения задания).

Кроме того, при подборе ресурсов Брокер ресурсов взаимодействует с каталогами подсистемы управления данными, чтобы найти местоположения файлов, указанных в списке входных данных JDL-файла. Это необходимо, чтобы направленное в грид задание по возможности выполнялось на рабочего узле, близком к используемым файлам.

WMS также взаимодействует с подсистемой безопасности и контроля прав, чтобы правильно идентифицировать пользователя и предоставить соответствующий доступ к ресурсам грида.

Наконец (но не в последнюю очередь) WMS взаимодействует с Вычислительными

элементами (CE) ресурсных центров для того, чтобы направить задания на выполнение, и для того, чтобы получать уведомления о состоянии ресурсов и их характеристиках. Упрощенная схема WMS и ее взаимодействия с другими подсистемами представлена на рис. 8.

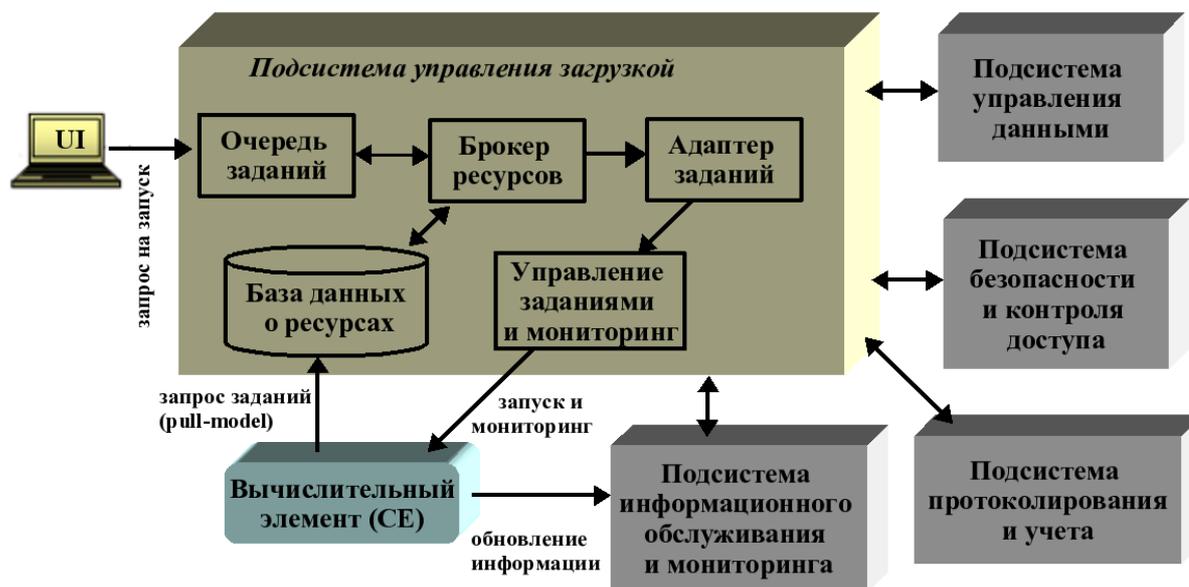


Рис. 8 Упрощенная схема подсистемы управления загрузкой и ее взаимодействия с другими компонентами грида

2.2.3 Вычислительный элемент

Вычислительный элемент (Computing Element, CE) это служба, представляющая вычислительные ресурсы данного сайта (Ресурсного центра) грида и выполняющая на нем функции управления заданиями (запуск, удаление и т.д.). Обращения к CE исходят либо непосредственно от интерфейса пользователя, либо от менеджера загрузки, который распределяет задания по множеству CE. Вообще говоря, CE может работать как в соответствии с моделью «нетерпеливого планирования» (push-модель), когда менеджер загрузки самостоятельно принимает решение о посылке задания на CE, так и в режиме «ленивого планирования» (pull-модель), когда CE запрашивает задание у WMS. Обе стратегии планирования имеют свои преимущества и недостатки. Выбор той или иной стратегии или их комбинации зависит от назначения и характеристик работы (в частности, интенсивности запуска заданий, числа ресурсных центров и т.п.) грид-системы.

Кроме того, CE содержит компоненты, являющиеся интерфейсом между высокоуровневым менеджером ресурсов и локальной системой управления вычислительными ресурсами грид-сайта (примерами таких систем являются PBS (Portable Batch System) [55], LSF (Load Sharing Facility) [56] или fork – простейшее стандартное средство запуска процессов в UNIX/Linux). Эти компоненты выполняют следующие функции:

- производят взаимную аутентификацию с клиентом;
- анализируют запрос (сделанный на языке описания заданий);
- отображает клиентский запрос на учетную запись некоторого локального (формального) пользователя, права которого соответствуют локальной политике ресурсного центра и правам грид-пользователя, запустившего задание;

- запускают задание и производят его дальнейший мониторинг, сообщая клиенту об ошибках, завершении задания и других событиях.

Помимо функций управления заданиями SE также поставляет информацию о состоянии ресурсов. В push-модели ее публикует информационная служба, и она используется WMS для выбора SE, на котором будет запускаться задание. В pull-модели эта информация пересылается WMS в сообщении, главным содержанием которого является: "SE доступен и готов принять новое задание".

2.3 Подсистема управления данными

Как правило, в грид-системах работают с данными на файловом уровне, в противоположность, например, системам баз данных, которые оперируют такими элементами как записи и поля. Подсистема управления данными (Data Management Subsystem, DM) включает три сервиса, поддерживающие доступ к файлам:

- ресурс хранения данных (Storage Element, SE),
- сервис каталогов (Catalog Services, CS),
- планировщик передачи данных (Data Scheduler, DS).

В распределенной грид-среде пользовательские файлы могут храниться во множестве экземпляров – реплик, размещенных в разных местах. Задача CS и DS состоит в том, чтобы сделать процесс управления репликами прозрачным для пользователя, так чтобы приложения получали доступ к файлам по их глобальным для всего грида именам или дескрипторам метаданных. Другими словами, хотя реально доступ к данным файлов происходит через один из множества SE, входящих в грид-систему, подсистема управления данными обеспечивает глобальную файловую систему в масштабах всего грида (концепция виртуализации наборов данных). Клиентское приложение для навигации по такой файловой системе может быть устроено как командная оболочка Unix/Linux - с использованием привычных команды смены каталогов, просмотра файлов и т.п. Защита файлов от несанкционированного доступа обеспечивается подсистемой безопасности и контроля прав доступа (раздел 2.5), в частности с помощью списков контроля доступа (Access Control List, ACL).

2.3.1 Наименование файлов в глобальном гриде

Прежде, чем обсуждать работу ресурсов хранения данных и каталогов подсистемы управления данными, необходимо ознакомиться с принципами наименования файлов в глобальной грид-системе.

Пользователь грид-системы идентифицирует файлы логическими именами файлов (LFN). Пространство имен LFN является иерархическим, точно так же как обычная файловая система. Семантика пространства имен LFN также практически такая же, как у файловой системы Unix/Linux. Итак, LFN - логическое имя файла, читаемый человеком идентификатор для файла, который является уникальным, но изменяемым: файлы могут быть переименованы пользователем. Каждая Виртуальная организация может иметь свое собственное пространство логических имен (если все придерживаются этого соглашения).

Однако LFN не единственное название/идентификатор, которое связано с файлом в гриде, хотя обычному пользователю другие имена файлов могут потребоваться лишь в крайне редких случаях. Примером формы LFN является следующая строка

```
/glite/myVO.org/production/run/07/123456/calibration/cal/cal-table100
```

LFN должен быть уникальным. В нашей архитектуре, интерфейсом грида, который используется, чтобы управлять LFN это каталог файлов и реплик, который, в частности, обеспечивает уникальность LFN. Если один и тот же каталог должен совместно использоваться несколькими Виртуальными организациями, то уникальность LFN должна быть обеспечена для всех этих ВО. Чтобы быть в состоянии совместно использовать каталог, ВО должны договориться о структуре имен для логической иерархии. Это может быть достигнуто с помощью префикса для каждой ВО. Тогда каждая ВО должен обеспечить уникальность LFN в пределах ее собственного пространства имен. Это тот же самый механизм, который используется в других распределенных файловых системах – таких как AFS.

В пространстве имен LFN определено понятие директории. Директориями можно управлять, как и в нормальной файловой системе: их можно просматривать, добавлять в них новые файлы и поддиректории или уничтожать/переименовывать уже существующие. LFN содержит полную иерархию директорий, которым это имя принадлежит. В предыдущем примере LFN, одна из директорий такова:

```
/glite/myvo.org/production/run/07/123456/calibration/
```

Таким образом, в пространстве логических имен (LFN) файлы представлены в виде обычной иерархической структуры (каталоги, подкаталоги и т.д.). Пользователи могут также копировать логические файлы из других логических директорий в их собственные директории.

В логическом пространстве имен файлов могут существовать символические ссылки (Logical Symlinks). Может существовать много Symlinks для данного LFN (отношение N:1). Если LFN перемещен или переименован, Symlink остается «повисшей», - по аналогии с обычной семантикой файловой системы. Напротив, для логических *директорий* символические связи, как правило, не поддерживаются из-за трудности отслеживания символических директорий в среде распределенных пространств имен и каталогов.

Чтобы поддерживать систему LFN, программные средства грида должны отображать логические имена на физические, обеспечивающие глобальную однозначную идентификацию файла, с указанием на его местоположение и протокол вызова. Такими физическими именами являются (см. рис. 9):

- **GUID** (Grid Unique IDentifier) - глобальный уникальный идентификатор. Каждый LFN взаимно однозначно связан со своим GUID. GUID являются неизменными, то есть они не могут быть изменены пользователем. Эти имена используются приложениями грида как неизменные указатели на файлы. Если проводить аналогию с файловой системой Unix/Linux, GUID соответствует уникальному inode-числу файла. Взаимнооднозначная связь LFN и GUID означает, что жесткие ссылки в виртуальной файловой системе грида не разрешены (в отличие от Unix/Linux) поскольку осуществление глобально распределенной файловой системы с жесткими ссылками является весьма трудной задачей. Заметим, что в то время как совокупность LFN имеет иерархическую структуру, GUID не имеют никакой структуры вообще.
- **SURL** (Site Universal Resource Locator; Site URL) - определяет физическое местоположение файла или его реплики. Файл может иметь много реплик - таким образом, отображение между GUID и SURL - "один ко многим". Каждая реплика файла имеет его собственный уникальный SURL. В качестве SURL выступает полное имя SRM, понятное интерфейсу SRM элемента хранения данных (SE). Пример SURL:

srm://castorgrid.cern.ch:8443/srm/managerv1?SFN=/castor/cern.ch/file1

Имя SRM неявно дается частью SURL, которая стоит перед символами ?SFN.

- **TURL** (Transport URL) - URL, который может использоваться, чтобы фактически передать файл, используя любой стандартный транспортный протокол. TURL начинается с протокола, используемого для передачи файла или прямого доступа к файлу через некоторый механизм ввода - вывода.

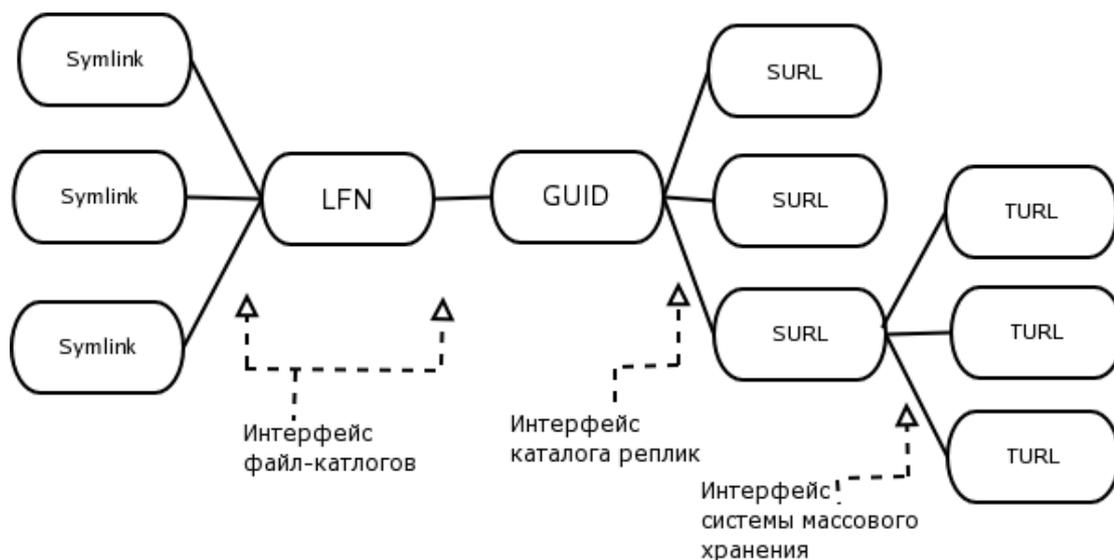


Рис. 9 Схема отображений различных пространств имен файлов в грид-системах

2.3.2 Ресурсы хранения данных

Прикладное задание должно быть в состоянии обратиться к своим данным, независимо от фактического местоположения процессора, на котором оно выполняется. При этом необходимо обеспечить доступ к системам хранения разного типа, существующим в ресурсных центрах – от простых файловых систем до устройств массового хранения данных. Чтобы не иметь дело с особенностями каждого индивидуального устройства хранения данных, каждый, включенный в грид-инфраструктуру ресурс хранения должен реализовывать единый интерфейс. В настоящее время наиболее распространенным и перспективным стандартом для такого интерфейса является Менеджер ресурсов хранения данных (Storage Resource Manager, SRM) [48], который предоставляет большинство функциональных возможностей, необходимых для управления данными в гриде. Интерфейс SRM, который принят в грид-системах, детально описан в документах, доступных на веб-сайте Глобального грид-форума (OGF, Grid Storage Management Working Group) [23].

Совокупность служб, необходимых для обеспечения доступа к файлам, хранящимся в каждом из ресурсных центров, называется Ресурс хранения данных (Storage Element, SE) грид-системы и состоит из:

- собственно устройства хранения со всеми соответствующими аппаратными средствами и драйверами;
- реализации интерфейса SRM;

- службы передачи данных для набора транспортных протоколов;
- службы ввода/вывода файлов (I/O);
- модулей взаимодействия с подсистемами регистрации и безопасности.

Непосредственно пользовательские прикладные задания во время исполнения на рабочем узле ресурсного центра используют службу ввода/вывода, чтобы обратиться к своим данным на устройстве хранения; SRM и службы безопасности используются косвенно.

Служба передачи файлов (File Transfer service, FTS) управляет входящими в SE потоками данных. Она аналогична службе управления пакетными заданиями вычислительной фермы. Ресурс хранения, на который передаются данные, всегда является локальным для данного экземпляра службы передачи, а ресурс хранения - источник данных может быть внутри или вне границ данного ресурсного центра. FTS функционально является довольно сложной и может состоять из нескольких компонентов. Например, в ППО gLite [22] служба передачи состоит из следующих компонентов:

- Служба размещения файлов - хранит все записи о передаче данных в своей базе данных и получает новые запросы. Пользователи взаимодействуют со службой передачи исключительно через клиентов этой службы.
- Агент передачи - опрашивает базу данных службы размещения о доступных передачах, для которых ресурс хранения, управляемый этим агентом, является пунктом назначения.
- Библиотека передач файлов - обеспечивает уровень управления поверх специализированных для грид протоколов передачи данных на большие расстояния. Она может контролировать состояние передачи и имеет возможность отменить ее.

Служба ввода-вывода принимает в качестве входных параметров либо LFN, либо GUID, сверяется со службой прав допуска к файлам - разрешен ли пользователю доступ к файлу, сопоставляет GUID или LFN с SURL, который затем используется, чтобы получить доступ к требуемому файлу.

2.3.3 Каталоги

Создание глобальной файловой системы на основе известных решений для ресурсов, распределенных в масштабе одной или нескольких тесно связанных организаций, типа AFS [49] в рамках глобального грида практически неосуществимо (или, во всяком случае, налагает слишком жесткие условия на провайдеров ресурсов). Вместо этого используются глобальные каталоги, позволяющие искать файлы и управлять файлами и их репликами в масштабах всего грида, создавая тем самым виртуальную файловую систему. Концептуально, различают Каталог метаданных (Metadata Catalog, MC), Файловый каталог (File Catalog, FC), Каталог реплик (Replica Catalog, RC) и Объединенный Каталог. Каталоги могут существовать как в виде локализованной, так распределенной версиях.

Каталог метаданных обеспечивает операции записи и получения результатов запросов по метаданным. Метаданные это данные о данных, то есть они должны на что-то ссылаться, о чем имеют дополнительную информацию. В данном случае метаданные ссылаются на логические имена файлов.

Метаданные сильно зависят от специфики прикладной области. Поэтому общий каталог должен содержать весьма ограниченный набор метаданных, но иметь интерфейсы к каталогам метаданных конкретных виртуальных организаций.

Файловый каталог предоставляет возможность производить действия над логическими

именами файлов. Такими действиями могут быть создание директорий, переименование файлов и создание символических ссылок.

Каталог реплик обеспечивает операции, связанные с дублированием (репликацией) файлов в гриде. В частности, он обеспечивает перечисление, добавление и удаление копий (реплик) файлов на разных ресурсах хранения данных, идентифицированных с помощью их GUID/SURL.

Таким образом, интерфейс RC дает доступ к отображению GUID - SURL, в то время как FC управляет только LFN, внутренне сохраняя соотношение с GUID.

Объединенный каталог обеспечивает все операции, которые нуждаются в синхронизированном доступе по крайней мере к двум каталогам, типа создания виртуальных (логических) директорий, создание и удаление новых файлов.

Каталоги должны поддерживать массовые операции с файлами (операции над совокупностью файлов) как часть своего интерфейса. Такие операции увеличивают производительность и оптимизируют взаимодействие пользователей с грид-службами.

Ресурсы хранения данных использует службы каталога, чтобы получить соответствие GUID и LFN с SURL и проверять права доступа к файлам.

2.4 Подсистема информационного обслуживания и мониторинга грида

Подсистема информационного обслуживания и мониторинга решает задачу сбора и управления данными о состоянии грида, получая информацию от множества распределенных источников – поставщиков. Подсистема предназначена для постоянного контроля функционирования грид-системы и обеспечения своевременного реагирования на возникающие проблемы.

Информационные системы общего назначения (базы данных и сервисы директорий) плохо подходят для распределенного мониторинга, ввиду природы самих данных. Статусные данные мониторинга имеют ограниченное и, как правило, короткое время жизни (после чего они становится недостоверными). Поэтому частота их обновлений должна быть высокой, в то время как обычные БД оптимизируются на запросы, а не на обновления. В информационной системе мониторинга должна обеспечиваться низкая задержка при передаче от точки получения данных к точке, где они хранятся. В свою очередь, принимающая сторона должна выдерживать высокую скорость приема, обусловленную частыми обновлениями.

Архитектура с такими свойствами (Grid Monitoring Architecture, GMA) предложена в [53]. Суть предложения заключается в том, чтобы разделить сбор данных и операции поиска. Данные мониторинга хранятся распределенно – там же где и производятся. Так как суммарный объем данных очень большой, задержки при поиске по всему информационному массиву будут непредсказуемы. Поэтому предлагается адресовать поисковые запросы “реестру матаданных”, который представляет собой индекс распределенного хранения и позволяет определить источник требуемых данных. Далее запрос переадресуется в место хранения и там производится уже более узкий поиск. Имеется реализация этого подхода (R-GMA [54]).

2.4.1 Архитектура и реализация

GMA определяет модель поставщик-потребитель, которая моделирует информационную инфраструктуру грида как множество, включающее:

- потребителей, которые запрашивают информацию,

- поставщиков, которые производят информацию,
- центральный реестр, который служит связующим звеном между поставщиками и потребителями.

Поставщики входят в контакт с реестром, чтобы объявить о намерении опубликовать данные, а потребители - чтобы идентифицировать поставщиков, которые могут обеспечить их требуемыми данными. Сами данные приходят потребителю непосредственно от поставщика, минуя реестр. Поскольку схема является достаточно общей, она применима как для хранения данных о гриде (какие ресурсы и сервисы доступны, каковы их характеристики), так и для мониторинга выполнения заданий.

Одной из наиболее распространенных систем грид-мониторинга является R-GMA [54], которая представляет собой реляционную реализацию GMA. При наличии множества распределенных поставщиков, с точки зрения информационных запросов R-GMA действует как одна большая реляционная база данных. "Реляционность" проявляется в форме представления данных: R-GMA добавляет стандартный язык запросов (подмножество языка структурированных запросов (Structured Query Language, SQL), который является стандартом ANSI/ISO [57]) к модели GMA. Поставщики объявляют о составе публикуемой информации посредством конструкции SQL CREATE TABLE, публикуют ее посредством SQL INSERT, а потребители получают данные через SQL SELECT. R-GMA также гарантирует, что все кортежи (строки таблиц в базах данных) имеют временную метку, что необходимо для поддержки систем мониторинга (которые требуют упорядоченных по времени данных). Основным способом получения информации является нотификация о событиях, происходящих в системе (push-модель).

R-GMA представляет информационные ресурсы какой-либо Виртуальной Организации (VO) в виде единой - хотя территориально-распределенной - базы данных. Единая схема базы данных содержит название и структуру (названия столбцов, типы и параметры настройки) каждой виртуальной таблицы в системе. Единая служба реестра для каждой таблицы содержит список (реестр) поставщиков, которые публикуют строки (предоставляют данные) для данной таблицы. Когда потребитель выполняет SQL-запрос, касающийся какой-либо таблицы, служба регистрации, ведущая реестр, выбирает лучших поставщиков, чтобы ответить на запрос. Этот процесс называется посредничеством. Затем потребитель входит в непосредственный контакт с каждым поставщиком, объединяет полученную информацию, и возвращает пользователю ряд кортежей. Процесс посредничества скрыт от пользователя. Заметим, что не существует никакого центрального архива для хранения виртуальных таблиц - именно в этом смысле база данных виртуальна.

С точки зрения потребителя существуют четыре типа запросов к поставщикам:

- непрерывный: предоставляет клиенту все результаты, соответствующие запросу, как только они опубликованы;
- последний: используемый, чтобы найти текущее значение какого-либо параметра;
- исторический: более традиционный запрос о происшедших событиях и значениях параметров за некоторый период времени (включая "все время");
- статический: запросы, типичные для обычных баз данных, они не содержат временной отметки R-GMA.

2.4.2 Структура поставщиков и потребителей подсистемы информационного обслуживания и мониторинга на основе R-GMA

R-GMA имеет четыре основных компонента, работающие на одном или более серверах:

- поставщики;
- потребители;
- схема данных;
- сервис реестра.

Сервис реестра хранит информацию обо всех доступных в данный период времени поставщиках информации. Сервис регистрации записывает и предоставляет информацию о поставщиках, которая включает описание и способ представления публикуемых данных, но не самих данных. Чтобы избежать узкого места и единственного пункта отказа, существует возможность поддержки многих копий регистрационных записей, которые достаточно часто синхронизируются. Описание данных фактически хранится как ссылка на таблицу, соответствующую схеме данных.

По способу формирования информации (кортежей для таблиц) поставщики подразделяются на три класса:

- первичный поставщик,
- вторичный поставщик,
- поставщик по требованию.

Основное различие между ними состоит в том, как формируются кортежи: первичные поставщики сами формируют информацию, хранят ее в своей памяти и отвечают на запросы потребителей; вторичные поставщики собирают информацию от первичных, хранят ее и отвечают на запросы; у поставщиков по требованию нет внутренней памяти - данные формируются непосредственно в ответ на запрос.

А с точки зрения обслуживаемых запросов (предоставляемой информации) поставщики подразделяются на пять типов:

- **DataBaseProducer**: отвечает на запросы по истории работы системы (для этого информация, формируемая поставщиком, записывается в базу данных);
- **StreamProducer**: поддерживает непрерывные запросы о текущем состоянии системы;
- **ResilientProducer**: подобен **StreamProducer**, но информация записывается на диск так, чтобы никакая информация не была потеряна в случае неполадок системы.
- **LatestProducer**: поддерживает последние запросы, сохраняя только последние записи в базе данных;
- **CanonicalProducer**: использует пользовательский код для ответа на SQL-запросы.

Другими важными компонентами R-GMA являются:

- **Archiver** – собирает определенную информацию по заданию пользователя, которую затем по соответствующему запросу ему отправляет;
- **Mediator** - компонент, благодаря которому система становится подобной виртуальной базе данных: когда запрос сделан, **Mediator** использует сервис регистрации, чтобы найти правильных поставщиков и затем комбинирует

информацию от них.

2.4.3 Период хранения кортежей

Первичные и вторичные поставщики периодически производят чистку своей памяти, удаляя «старые» кортежи. Точное значение понятий «старый кортеж», и «текущее состояние» (для запросов типа «последний») определяется с помощью двух типов периода хранения кортежей:

LatestRetentionPeriod (период хранения последних кортежей) определяет – после какого времени хранения кортеж не может рассматриваться как последний. Этот интервал времени добавляется к временной метке и вставляется в каждый кортеж, публикуемый первичным поставщиком, и остается там, когда кортеж заново публикуется вторичным поставщиком;

HistoryRetentionPeriod (период хранения истории) - объявляются первичными и вторичными поставщиками для каждой таблицы, в которой они публикуют кортежи.

Таким образом, первичные и вторичные поставщики имеют два логических набора кортежей: один для запросов типа «последний», а другой – для непрерывных и исторических запросов. Поставщики обязаны сохранять новую версию любого кортежа, который не превысил LatestRetentionPeriod, и все версии любого кортежа, которые не превысили HistoryRetentionPeriod.

Все первичные и вторичные поставщики поддерживают непрерывные запросы, однако не все из них обязательно поддерживают исторические и последние запросы.

HistoryRetentionPeriod может быть длиннее или короче, чем LatestRetentionPeriod. HistoryRetentionPeriod - свойство поставщиков, тогда как LatestRetentionPeriod - свойство кортежа.

Для работы с R-GMA пользователь может использовать интерфейс командной строки или специальный браузер (графический интерфейс). Эти интерфейсы, в частности, позволяют:

- просматривать схему данных в R-GMA (определения доступных таблиц);
- просматривать списки поставщиков, которые публикуют данные в таблице;
- направлять запросы к таблицам через медиатор;
- направлять запросы к определенным поставщикам таблицы.

Чтобы использовать браузер R-GMA, когда он работает на безопасном (HTTPS) сервере, пользователь должен использовать специальный электронный сертификат (см. следующий раздел 2.5), который должен быть импортирован в веб-браузер.

2.5 Подсистема безопасности и контроля прав доступа

Подсистема безопасности обеспечивает безопасный доступ к ресурсам в незащищенных сетях общего доступа (Интернет) с учетом прав данного пользователя и правил обслуживания пользователей данным ресурсным центром (такие правила часто называют «локальной политикой»). Практически во всех крупных грид-системах работа этой подсистемы основана на инфраструктуре безопасности грида (Grid Security Infrastructure, GSI), которая разработана Globus Alliance [19]. Подсистема предоставляет такие сервисы, как аутентификация, конфиденциальность передачи информации и делегирование прав. Под делегированием прав подразумевается, что пользователю нужно лишь один раз пройти процедуру аутентификации, а далее система сама позаботится о том, чтобы аутентифицировать его на всех ресурсах, которыми он собирается воспользоваться. GSI, в свою очередь, основана на надежной и широко используемой технологии открытых

криптографических ключей (Public Key Infrastructure, PKI).

2.5.1 Некоторые термины и общие принципы алгоритмов шифрования

В этом разделе приведены самые минимальные сведения о способах аутентификации и шифрования, необходимые для понимания дальнейшего материала. Более подробные сведения можно найти в обширной литературе по криптографии и способам защиты информации в компьютерных сетях, например в [60], [61].

Алгоритм шифрования это набор действий, необходимый для шифрования/дешифровки данных. Введем некоторые основные криптографические понятия:

- ключ – параметр алгоритма шифрования;
- аутентификация – проверка подлинности объекта (пользователя или грид-узла), направившего запрос на выполнение какого-либо действия;
- авторизация – сопоставление объекта и набора прав (привилегий) при работе в какой-либо системе (в нашем случае – в грид-системе);
- конфиденциальность – доступность каких-либо данных только заранее определенному набору объектов;
- целостность – неизменность передаваемых данных;
- цифровая подпись – инструмент для идентификации источника данных.

Алгоритмы шифрования могут быть симметричными и несимметричными. Их схемы представлены на рис. 10. Поскольку несимметричные алгоритмы гораздо медленнее симметричных (примерно в 1000 раз), для шифрования данных применяются гибридные технологии: несимметричный алгоритм применяется для начального согласования параметров шифрования и безопасной передачи симметричного ключа, а симметричный алгоритм для шифрования данных.

В подсистеме безопасности и контроля прав доступа в основном используется несимметричный алгоритм. При этом в режиме шифрования открытый ключ используется для шифрования, закрытый ключ – для расшифровки. В режиме цифровой подписи: закрытый ключ – для шифрования, открытый ключ – для расшифровки. Цифровую подпись может создать только носитель закрытого ключа. Поэтому важным требованием является безопасное хранение закрытого ключа.

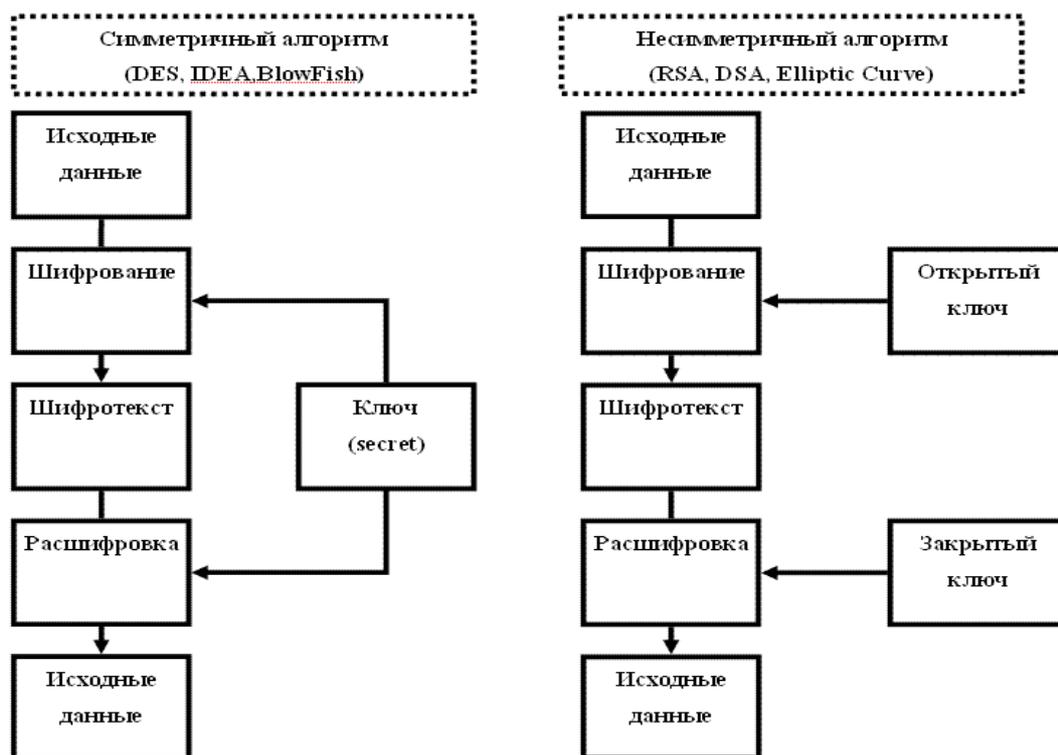


Рис.10 Алгоритмы шифрования

Рис. 10 Алгоритмы шифрования

2.5.2 Идентификация пользователей и узлов грида

В качестве идентификаторов пользователей и ресурсов в GSI используются цифровые сертификаты стандарта X.509 [58] (стандарт международной организации International Telecommunication Union, ITU [59]). В работе с сертификатами X.509 и в процедуре выдачи/получения сертификатов задействованы три стороны:

- Центр Сертификации (Certificate Authority, CA) – специальная организация, обладающая полномочиями выдавать (подписывать электронным способом) цифровые сертификаты. Различные CA обычно независимы между собой. Отношения между CA и его клиентами регулируются специальным документом и доверие сертификату строится на доверии центру сертификации, подписавшему этот сертификат.
- Владелец сертификата - пользователь грида или грид-ресурс, который пользуется сертификационными услугами CA. CA включает в сертификат данные, предоставляемые владельцем (имя, организация и другую информацию) и заверяет его своей цифровой подписью. В частности владельцу сертификату присваивается уникальное имя (Distinguished Name, DN).
- Пользователи или ресурсы, проводящие аутентификацию других грид-объектов – они полагаются на информацию из сертификатов при получении его от идентифицируемых объектов. Они могут принимать или отвергать сертификаты, подписанные какой-либо CA.

В GSI используются два типа сертификатов X.509:

- Сертификат пользователя (User Certificate) должен иметь каждый пользователь, работающий с грид-системой. Сертификат пользователя содержит информацию об имени пользователя, организации, к которой он принадлежит, и центре сертификации, выдавшем данный сертификат.
- Сертификат узла (Host Certificate) должен иметь каждый узел (грид-сервис или ресурс) грид-системы. Сертификат узла аналогичен сертификату пользователя, но в нем вместо имени пользователя указывается доменное имя конкретного грид-узла.

Сертификаты стандарта X.509 содержат открытый ключ и некоторые другие данные о пользователе или грид-узле. Действительные сертификаты хранятся в специальном репозитории центра сертификации; там же хранится список отозванных сертификатов (Certificate Revocation List, CRL). Сертификационный центр удостоверяет принадлежность сертификата данному пользователю или грид-узлу, которые определяются своими уникальными именами (Distinguished Name, DN).

Аутентификация в PKI сводится к следующим шагам:

- Объект (пользователь, грид-узел) А хочет аутентифицировать объект Б.
- Б посылает свой сертификат А, он проверяет правильность сертификата и подпись (или цепочку подписей) сертификационного центра.
- А посылает Б произвольную фразу с просьбой зашифровать её закрытым ключом Б.
- Б шифрует пришедшие данные и отправляет ответ А.
- А расшифровывает ответ Б с помощью переданного ранее открытого ключа и сравнивает результат с эталонной фразой.
- Если сравнение успешно, то Б действительно владеет закрытым ключом, соответствующим сертификату.

Владелец сертификата – может использовать закрытый ключ и сертификат для шифрования и цифровой подписи. Цифровая подпись это хэш данных зашифрованный закрытым ключом (напомним, что хэш – это число, которое ставится в соответствие данным таким образом, чтобы вероятность появления различных данных с одинаковым хэшем была близка к нулю, а восстановить данные по их хэшу было практически невозможно). Подпись может быть проверена с помощью открытого ключа.

2.5.3 Делегирование прав и использование доверенностей

Для обеспечения безопасности в ГРИД-системах используются не сами пользовательские сертификаты, а специальные доверенности (проху certificate, прокси-сертификат), с коротким (порядка нескольких часов) сроком действия. Эти доверенности выписывает сам пользователь с помощью своего постоянного сертификата, действуя в этом случае как «сертификационный центр» для самого себя. С их помощью грид-сервисы выполняют действия от лица пользователя – владельца сертификата, например запускают задания на вычислительном ресурсе.

Важным условием для эффективной работы распределенных систем является возможность делегирования прав пользователя грид-сервисам. Дело в том, что практически любой запрос пользователя проходит через несколько сервисов. И если бы не было механизма делегирования, пользователю было бы необходимо аутентифицироваться на каждом сервисе в цепочке, обрабатывающих данный запрос. Это означает, что пользователь после отправления задания должен неотрывно быть около своего компьютера и отвечать на запросы о своей аутентификации от каждого сервиса в обрабатывающей цепочке. Фактически это делает невозможным работу в грид-среде – во

всяком случае, при запуске большого набора заданий.

Делегирование прав позволяет избежать этой проблемы. Процесс делегации заключается в следующих шагах:

- сервис, которому делегируются права (делегат) создает пару ключей (открытый и закрытый);
- открытый ключ отсылается делегирующему (то есть пользователю – обладателю исходного сертификата или предыдущему в цепочке сервису);
- подписанный открытый ключ (сертификат) возвращается делегату вместе со всей цепочкой сертификации.

Это обеспечивает то, что пользователю достаточно выполнить аутентификацию только на первом сервисе в обрабатывающей запрос цепочке.

Прокси-сертификат не может быть отозван, поэтому создание долгосрочной доверенности очень нежелательно. Однако, при выполнении операций, требующих достаточно большого промежутка времени, может истечь срок действия такого доверенности, что приведет к некорректному завершению операции. Чтобы исключить подобные случаи используют специальный сервис, который по запросу пользователя автоматически возобновляет срок действия прокси-сертификата без вмешательства пользователя. За своевременным обновлением прокси-сертификата следит брокер ресурсов подсистемы управления загрузкой. При использовании сервиса возобновления доверенностей, пользователь целикомверяет ему свои полномочия на все время действия долговременного сертификата.

2.5.4 Сервис управления виртуальными организациями и авторизация пользователей

Как уже отмечалось в первой части, виртуальная организация (ВО) является одним из ключевых понятий грид-технологий. ВО объединяют сообщество пользователей, совместно работающих в некоторой прикладной области. Таким образом, пользователи, принадлежащие одной ВО, предъявляют к грид-ресурсам приблизительно однородные требования и могут быть обслужены на унифицированном уровне, что значительно упрощает управление гридом. Каждая ВО имеет собственную политику, и в соответствии с ней предоставляются ресурсы пользователям данной ВО.

С технической точки зрения сведения о зарегистрированных в данной грид-инфраструктуре виртуальных организациях и их составе хранятся в базе данных, которая используется ресурсными центрами для решения вопроса о допуске заданий конкретного пользователя на данный ресурс. Грид-ресурсы предоставляются на основе политики самой виртуальной организации и локальной политики ресурсных центров – в этом и состоит процесс авторизации. Для гибкого управления правами различных пользователей виртуальная организация может иметь внутреннюю структуру – содержать различные группы пользователей, а отдельным пользователям могут быть приписаны различные роли. Этим различным группам и ролям в процессе авторизации сопоставляются различные права доступа к грид-ресурсам (в соответствии с политикой ресурсных центров).

В ППО gLite информацию для авторизации пользователей предоставляет сервис VOMS (Virtual Organization Membership Service – Сервис управления членством в виртуальных организациях). Сервер VOMS использует реляционную базу данных MySQL или ORACLE в качестве систем хранения данных и основывается на добавлении некритичных расширений к пользовательскому прокси-сертификату, которые и содержат сведения о пользователе, необходимые для его авторизации.

2.6 Подсистема протоколирования

Подсистема протоколирования (Logging and Bookkeeping, LB) отслеживает процесс выполнения заданий, управляемый подсистемой управления загрузкой (WMS), отслеживает выполняющиеся в разных узлах грид-системы шаги обработки заданий, фиксирует происходящие с ними события (запуск, распределение в подходящий ресурсный центр, начало выполнения и так далее) и запоминает их. Она собирает извещения о событиях от различных компонентов WMS и обрабатывает их, чтобы представить обобщенное текущее состояние (статус) задания.

Пользователь не участвует в сборе данных от компонентов WMS и просто обращается к LB за требуемой информацией. LB предоставляет интерфейсы для запросов информации о заданиях, а также для регистрации запросов на получение уведомлений об изменениях состояния заданий.

2.6.1 Типы запросов к подсистеме протоколирования

Подсистема протоколирования поддерживает два типа запросов - запросы о состоянии заданий, которые возвращают детальное описание состояний одного или более заданий, и запросы о событиях, которые возвращают информацию о событиях, получаемую LB от компонентов WMS. Как правило, запросы о заданиях используются, чтобы проследить штатную обработку заданий; запросы о событиях используются, главным образом, для того, чтобы проследить аварийное поведение.

Каждый запрос содержит нескольких условий. Например, указывается конкретный идентификационный номер задания (он присваивается заданию подсистемой загрузки и сообщается пользователю, направившему это задание), имя владельца задания (как оно указано в сертификате владельца), специфическое состояние задания и так далее. Подсистема протоколирования преобразует условия в сообщение запроса, обрабатывает его и ждет ответа от соответствующей компоненты грида, который затем передается пользователю синхронным образом.

Другой способ взаимодействия пользователей с подсистемой протоколирования - зарегистрироваться для получения уведомлений. Их поставляют слушающему клиенту асинхронно, когда происходит определенное событие (обычно - изменение состояния задания). Основная цель этой функциональной возможности подсистемы протоколирования - избежать ненужной загрузки сервера LB многочисленными повторными запросами, причем в большинстве случаев - с одинаковым результатом.

Используя клиента уведомлений, пользователь регистрируется на сервере LB, для получения уведомлений. При этом он должен определить условия, при которых уведомление посылаются. Запрос на регистрацию посылается серверу LB таким же образом, как синхронные запросы, и сохраняется там. В ответе, сервер указывает уникальный идентификационный номер запроса на уведомления, с помощью которого пользователь в дальнейшем может обращаться к этому серверу, например для изменения условий, которые вызывают уведомление, продления периода действия регистрации или ее отмены и даже для изменения адресата уведомлений.

2.6.2 Безопасность и управление доступом к информации подсистемы протоколирования

Информация о заданиях, хранящаяся на сервере LB, должна быть доступна только для владельца задания (пользователя, который отправлял задание), и, возможно, для тех пользователей, которых владелец указал в специальном списке контроля доступа (Access Control List, ACL). Пользователи в ACL могут быть определены непосредственно по их именам (указанным в их сертификатах), или по названиям групп виртуальных

организаций (или по названиям целых ВО).

Для обеспечения контроля за доступом к информации все компоненты LB должны взаимодействовать на основе взаимной аутентификации, а пользователи, делающие запрос на сервер LB должны иметь действительный сертификат. Все сообщения, посланные по сети, шифруются, чтобы их содержание было недоступно для посторонних.

2.6.3 Источники извещений о событиях

В заключение этого раздела приведем список возможных источников извещений, связанных с выполнением заданий в грид-системе:

- Пользовательский интерфейс регистрирует задание в LB и предоставляет информацию относительно передачи задания ресурс-брокеру.
- Брокер ресурсов регистрирует различные события по мере прохождения задания через компоненты WMS, а также другую важную информацию, связанную с заданием (например, выбор CE, на котором будет выполняться задание).
- Вычислительный элемент обеспечивает информацию о ходе выполнения задания.
- Помимо указанных выше компонентов WMS, генерировать события может также специальный фрагмент – тэг (User Tag) - пользовательского кода в самом задании.
- Задания с контрольными точками (checkpointable) также могут использовать LB, чтобы следить за продвижением задания.
- Наконец, сама LB может быть источником извещений, например, при изменении списков управления доступом к информации о задании (ACL).

2.7 Подсистема учета

До настоящего времени существующие глобальные грид-системы использовались для научных исследований, в которых ресурсы предоставлялись научно-исследовательскими организациями в совместное использование для достижения общих некоммерческих целей. Причем эти грид-системы, в основном, работали в тестовом режиме. Поэтому детальный учет использования ресурсов не был первоочередной задачей.

Однако с переходом грид-систем к полнофункциональному обслуживанию пользователей из различных областей вопрос об учете использования грид-ресурсов становится весьма актуальным – в том числе, и в связи с возможным введением оплаты (в той или иной форме) за использование ресурсов.

Подсистема учета (Accounting Subsystem, AS) аккумулирует информацию об использовании грид-ресурсов отдельными пользователями, группами пользователей и виртуальными организациями. На основе собранной информации может формироваться политика распределения ресурсов и взиматься плата за их использование. Кроме того, эта информация позволяет осуществлять подготовку статистических сообщений, проследивать использование ресурса индивидуальными пользователями. Она может также использоваться, чтобы осуществлять политику доступа, основанную на пользовательских квотах на использование ресурсов, а также для улучшения балансировки загрузки грид-ресурсов.

Мы кратко поясним принципы работы этой подсистемы на примере ППО gLite.

2.7.1 Подсистема учета ППО gLite

Программное обеспечение подсистемы учета DGAS (DataGrid Accounting System) ППО gLite не имеет центрального архива учетной информации. Вместо этого такая информация распределена по набору независимых серверов учета, которые ведут записи учета групп пользователей и ресурсов грида.

Подсистема учета состоит из трех компонентов:

- службы регистрации пользователей и ресурсов, а также хранения учетной информации;
- службы формирования цены;
- службы сбора информации об использовании ресурсов.

Постоянно работающие агенты службы сбора информации об использовании ресурсов, устанавливаются в ресурсных центрах - на вычислительные элементы (CE) и элементы хранения данных (SE).

Служба регистрации пользователей и ресурсов и хранения учетной информации (Home Location Register, HLR) является ответственной за хранение учетной информации для пользователей и для ресурсов грида. Она получает учетную информацию - так называемые, Отчеты об использовании, - от службы сбора информации и хранит ее для обслуживания последующих запросов. Информация, полученная от HLR, может быть отсортирована по пользователям, ресурсам, или по выполненным заданиям.

Эти отчеты использования являются основой для последующего подсчета (совместно со службой формирования цены, см. ниже) стоимости задания и для возможных взаиморасчетов пользователей и провайдеров ресурсов. Кроме общей информации об учетных записях пользователей и ресурсов, сервер HLR хранит информацию об использовании ресурсов каждым из заданий, связанных с учетными записями ресурса и/или пользователя.

Существуют два типа серверов HLR:

- пользовательские HLR хранят информацию с точки зрения пользователей и предоставляют пользователям информацию о них самих и о заданиях, которые они направляли в грид;
- ресурсные HLR хранят информацию об использовании ресурсов и обслуживают запросы администраторов или владельцев ресурсных центров.

Это сделано для масштабируемости подсистемы учета: с ростом числа пользователей грида, будет расти число пользовательских HLR, и различные пользователи будут зарегистрированы в различных HLR. При этом необходимо, чтобы вся отчетная информация относительно данного пользователя была отправлена и хранилась на (пользовательском) HLR, который управляет его учетной записью. С другой стороны SE получает задания от различных пользователей, которые зарегистрированы во многих различных HLR. Ясно, что владелец ресурса, который нуждается в исчерпывающей информации об использовании его CE, не может делать запросы во все пользовательские HLR, чтобы ее получить (то же самое, конечно, справедливо для пользователей, которые не должны делать запросы в несколько ресурсных HLR). Следовательно, копии отчетов использования ресурсов всеми заданиями, выполняемыми на данном SE должны присутствовать на другом HLR, который управляет учетной записью CE ("ресурсный" HLR). Таким образом, владелец ресурса может сделать запрос только в единственный HLR, чтобы иметь исчерпывающее представление об использовании его ресурсов. Специальный компонент обеспечивает сопоставление отчетов использования с учетными

записями пользователя/ресурса, а также обмен сообщениями между пользовательским и ресурсным HLR.

Служба формирования цены (Price Authority, PA) назначает цену за использование ресурсов грида в пределах своего административного домена. Цены, которые хранятся в ценовой базе данных, могут быть назначаться вручную или с использованием различных динамических алгоритмов оценки. Цена ресурса используется, чтобы вычислить стоимость выполнения задания и последующих взаиморасчетов провайдеров и потребителей грид-ресурсов. Предлагается, что должен быть один сервер PA для каждой виртуальной организации.

На этом мы заканчиваем краткий обзор назначения и принципов работы основных компонентов глобальных грид-систем и переходим к описанию грид-инфраструктуры в целом. Для конкретности, мы рассмотрим крупнейшую в настоящее время грид-инфраструктуру, созданную в рамках проекта EGEE (Enabling Grids for E-sciencE) [13].

3 Реализация грид-технологий в проекте EGEE/LCG/RDIG

Соответствие всем трем критериям грида, указанным в разделе 1.4.2, наиболее четко прослеживается для глобальных систем, разрабатываемых для решения крупных научных задач. Примерами таких систем и грид-проектов являются: Open Science Grid [50], AliEn [51], Nordugrid [52] и EGEE [13]. В силу ограниченных размеров статьи, мы сосредоточимся на крупнейших из них – EGEE, в котором авторы – в составе грид-группы НИИЯФ МГУ - принимают непосредственное участие.

Инфраструктура EGEE создается на основе исследовательской сети Европейского союза (ЕС) GEANT и в ней использован опыт, накопленный в таких проектах, как DataGrid [16], LCG [12] и в национальных проектах, например, e-Science, INFN Grid, Nordugrid, и Open Science Grid. Она обеспечит возможность совместной работы с другими гридами во всем мире, включая США и Азию, что будет способствовать появлению всемирной грид-инфраструктуры.

3.1 Основные цели и структура проекта EGEE

Главной целью проекта EGEE (Enabling Grids for E-sciencE - Развёртывание Грид-систем для развития e-науки) является создание глобальной грид-инфраструктуры. В результате его осуществления, ученые, работающие как в академических институтах, так и в промышленности, получают доступ к значительным географически распределенным ресурсам вычислительной техники вне зависимости от того, где они находятся. По аналогии с аналогичными бизнес-приложениями (e-бизнес - бизнес в сети Интернет) исследования с использованием распределенных систем называли «e-наукой».

Более детально, задачами проекта являются:

- развертывание унифицированной, надежной, масштабируемой грид-системы и соответствующей инфраструктуры,
- наращивание ресурсов вычислительной техники и хранения данных, объединенных этой грид-инфраструктурой;
- совершенствование промежуточного программного обеспечения;
- привлечение новых пользователей как из различных научных, так и из производственных областей;
- обеспечение высокого уровня их обучения и поддержки при использовании грид-инфраструктуры проекта.

Проект финансируется Европейским Сообществом и странами-участниками. Его первый двухлетний этап закончился в марте 2006 года, после чего начался второй, тоже двухлетний, этап – EGEE-II. Цель EGEE-II – на основе результатов проекта EGEE создать полнофункциональную, постоянно работающую глобальную грид-инфраструктуру «производственного» уровня, взаимодействующую с другими гридами во всем мире. В результате появится исключительно высокопроизводительная всемирная инфраструктура, намного превосходящая по своим возможностям локальные кластеры и отдельные центры.

В консорциум EGEE-II входят свыше 90 участников из 32 стран. Они объединены в 12 федераций и представляют почти все основные европейские международные и национальные грид-проекты, а также проекты в США и Азии. Кроме того, несколько родственных проектов (48 ассоциированных участников) распространяют грид-инфраструктуру на Средиземноморье, Балтику, Латинскую Америку и Китай. Множество проектов, развившихся из EGEE и EGEE-II или связанных с ними, подтверждают роль

EGEE как инкубатора грид-технологий.

Грид-инфраструктура EGEE уже стала повседневным рабочим средством для целого ряда больших и малых исследовательских сообществ. В ней работают приложения для физики высоких энергий, биологических наук и смежных дисциплин, наук о Земле, астрофизики, вычислительной химии, термоядерной энергетики и других. Более подробно о прикладных областях, которые решают свои задачи с помощью грида EGEE будет рассказано в конце этой части.

3.1.1 Направления работ по проекту

В соответствии с задачами проекта, работа в EGEE-II подразделяется на следующие направления:

- обеспечение развития грид-сообщества (Networking Activities, NA);
- функционирование служб грид-инфраструктуры (Service Activities, SA);
- совместные исследования участников проекта (Joint Research Activities, JRA).

Первое направление (NA) включает в себя:

- NA1 – управление проектом;
- NA2 – распространение знаний и опыта использования грид-технологий;
- NA3 – обучение и подготовка пользователей грида;
- NA4 – поиск и взаимодействие с прикладными областями, в которых может с успехом использоваться грид, и их поддержка в грид-инфраструктуре;
- NA5 – выработка общей политики проекта и вопросы международного сотрудничества.

В рамках второго направления (SA) функционируют следующие службы:

- SA1 – поддержка, эксплуатация и управление грид-инфраструктурой;
- SA2 – обеспечение сетевыми ресурсами;
- SA3 – интеграция, тестирование и сертификация ППО; эта служба объединяет элементы промежуточного программного обеспечения из разных источников с целью получить интегрированные релизы, готовые к размещению в инфраструктуре проекта.

Совместные исследования (JRA) ведутся в следующих областях:

- JRA1 – развитие промежуточного программного обеспечения (разработка и поддержка gLite);
- JRA 2 – обеспечение качества работы грид-инфраструктуры; в частности, общая координация мер безопасности.

Большое внимание во второй фазе проекта уделяется поддержке приложений и распространению информации о проекте, чтобы обеспечить быстрое увеличение числа пользователей по сравнению с первой фазой EGEE. При этом особая важность придается привлечению приложений из производственных областей. Для этого разработана специальная программа EGEE по работе с бизнес-партнерами (Business Associate) и, в частности, организован Индустриальный форум (Industry Forum) – регулярная серия различных мероприятий (конференции, семинары, встречи), на которых представители индустрии могут ознакомиться с возможностями грида, а представители EGEE – с

нуждами и запросами прикладных задач в производственных областях. Важным каналом взаимодействия с индустрией является взаимодействие с проектом CERN Openlab, объединяющим ЦЕРН (Европейский центр ядерных исследований, CERN – базовую организацию всего проекта EGEE) и ведущие IT-фирмы.

3.1.2 Критерии успеха проекта

Успех грид-инфраструктуры EGEE будет оцениваться по показателям качества сервиса. Кроме количественной оценки того, насколько грид-технология повышает эффективность решения разных задач, внимание будет уделяться и качественным достижениям в плане функциональности, которая ранее не была доступна участвующим в проекте научным сообществам:

- **Упрощенный доступ.** EGEE уменьшит издержки, связанные с разнообразием не связанных между собой систем учета пользователей. Пользователи смогут объединяться в виртуальные организации с доступом к гриду, содержащему нужные каждому пользователю рабочие ресурсы.
- **Выполнение вычислений по требованию.** Эффективно распределяя ресурсы, грид-технология значительно сокращает время ожидания доступа к ним.
- **Географически распределенный доступ.** Инфраструктура будет доступна отовсюду, где обеспечена хорошая стыкуемость сетей. Ресурсы, таким образом, становятся более широко доступными.
- **Исключительно большой объем ресурсов.** Благодаря согласованности ресурсов и пользовательских групп, в прикладной работе в рамках EGEE будут доступны ресурсы таких объемов, какие не может предоставить ни один компьютерный центр.
- **Совместное использование программного обеспечения и данных.** Благодаря единой структуре вычислительных средств, в EGEE географически распределенным сообществам пользователей будет легко совместно использовать программное обеспечение и базы данных и разрабатывать программное обеспечение.
- **Высокий уровень поддержки приложений.** Компетентность всех участников EGEE обеспечит тщательную, всестороннюю поддержку всех основных приложений.

3.2 Компоненты грида EGEE

Как мы отмечали в первой части, грид - это соединение технологии (специальное программное обеспечение), инфраструктуры (аппаратных средств и грид-службы) и стандартов.

3.2.1 Технологии и стандарты: промежуточное программное обеспечение грида EGEE

Грид-система EGEE работает на основе промежуточного программного обеспечения gLite [22], созданного в рамках того же проекта (с использованием как оригинальных разработок, так и компонент ППО других грид-проектов). Наше описание принципов построения ППО в предыдущей части близко следовало общей архитектуре и функциональному назначению отдельных компонент gLite. Поэтому здесь мы не будем детально описывать это ППО, отсылая интересующихся к Веб-сайту [22], где можно найти полную информацию.

Как отмечалось, большая часть компонент этого ППО построена на основе Веб-сервисных

технологий и принципов OGSA (об OGSA рассказано в п. 1.4.3.2.8). Предполагается, что по мере становления других грид-стандартов ППО gLite будет эволюционировать таким образом, чтобы следовать этим стандартам. В частности, это касается спецификаций WSRF (п. 1.4.3.2.9) – если они получат широкое признание грид/веб-сообщества.

3.3 Инфраструктура

Инфраструктура грид состоит из грид-ресурсов, базовых сервисов, обеспечивающих предоставление этих ресурсов пользователям и ряда специальных служб, предназначенных для контроля за функционированием инфраструктуры.

В качестве грид-ресурсов обычно рассматриваются вычислительные ресурсы и ресурсы хранения данных.

Вычислительные ресурсы EGEE как правило представляют из себя кластеры, построенные на основе персональных компьютеров (ПК) и соединенных локальной сетью. В последнее время для этих целей, как правило, используются ПК, которые легко и компактно устанавливаются в специальные стойки или шкафы. Размер кластера может составлять до нескольких сотен ЦПУ. Вообще говоря, в грид-инфраструктуру могут быть включены суперкомпьютеры и другие специализированные вычислительные ресурсы, но EGEE основан на кластерах.

Важную роль в функционировании грид играют ресурсы хранения данных. Особенно важно это для приложений, которые связаны с интенсивной обработкой данных, например, приложения в физике высоких энергий. Такие ресурсы создаются на основе больших массивов на жестких дисках и/или роботизированных ленточных библиотек. Объем отдельного ресурса может достигать десятков и даже сотен терабайт.

Как правило, вычислительные ресурсы и ресурсы хранения объединяют локальной сетью в так называемый ресурсный центр. Это позволяет приблизить выполнение задач пользователя к данным и, тем самым, сократить трафик в сети. Как мы объясняли в предыдущей части, доступ к каждому типу ресурсов осуществляется через специальные интерфейсы: вычислительный элемент (Computer Element, CE) для вычислительного ресурса, элемент хранения (Storage Element, SE) для ресурса хранения данных.

Доступ пользователей к грид ресурсам осуществляется посредством пользовательского интерфейса.

К концу 2006 года инфраструктура EGEE включала около 200 ресурсных центров, порядка 25000 ЦПУ, а общий объем ресурсов хранения данных был свыше 5 Петабайт.

Представление о географии грид-среды EGEE дает рис. 11, а запуска заданий в ней - рис. 12. Фактически, это просто снимки с экрана компьютера, подключенного через веб-интерфейс к службе постоянного мониторинга этого грида (<http://gridportal.hep.ph.ic.ac.uk/rtn>). Точки на этой карте соответствуют ресурсным центрам и базовым грид-сервисам.

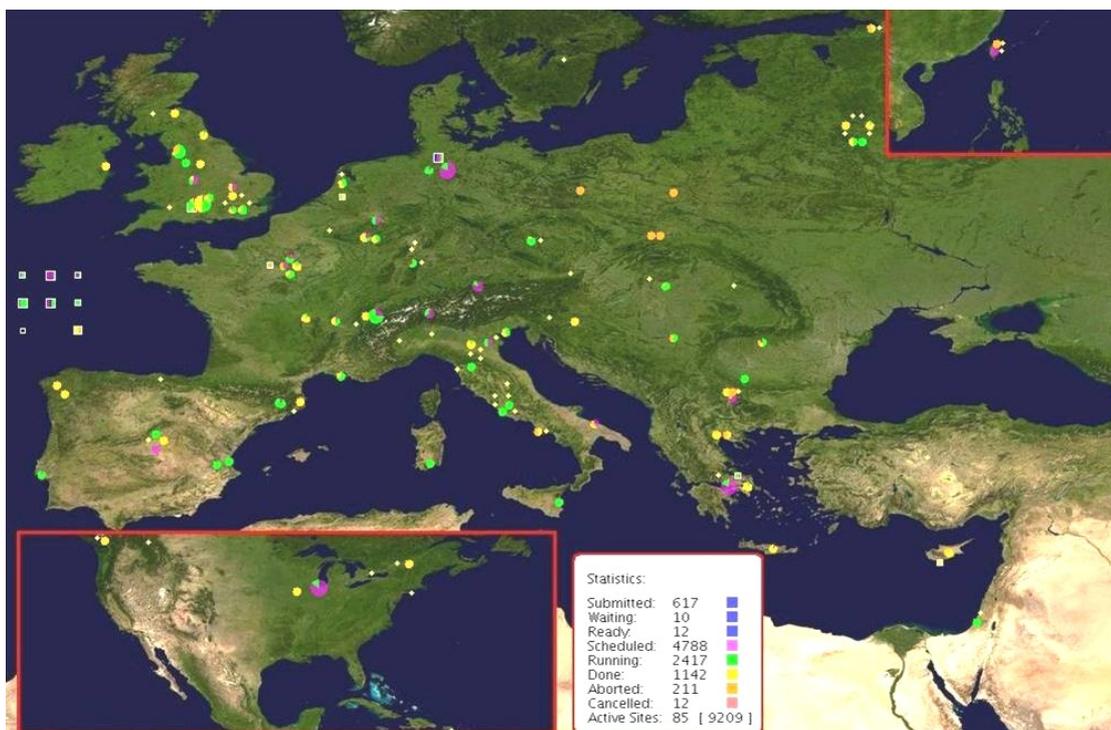


Рис. 11 Мгновенный снимок карты постоянного мониторинга грид-среды EGEE

Для устойчивой работы грида и комфортной работы пользователей в инфраструктуре действуют также различные общесистемные службы на основе людских ресурсов, а не ППО. Например, для бесперебойной работы всей грид-инфраструктуры EGEE и своевременного устранения возникающих проблем специальные команды, сформированные на базе региональных операционных центров, осуществляют постоянное дежурство, сменяя друг друга каждую неделю. Эти оперативные группы постоянно наблюдают за параметрами глобальной грид-системы и несут ответственность за бесперебойность ее работы. В России такая группа организована на базе НИИЯФ МГУ. Другим примером является служба поддержки пользователей, куда они могут обратиться с вопросами и за помощью в случае возникновения проблем при использовании грида EGEE. В России такая служба действует на базе ИТЭФ (<http://ussup.itep.ru>).

Сердцем грид-инфраструктуры являются базовые грид-службы (раздел 2.1). Работа разных базовых служб и разных экземпляров этих служб обеспечивается разными участниками EGEE – в соответствии с распределением обязанностей в рамках проекта. Следующий рисунок также является мгновенным снимком карты постоянного мониторинга работы системы, но на котором показано как брокеры ресурсов подсистемы распределения заданий распределяют в данный момент задания по ресурсным центрам.

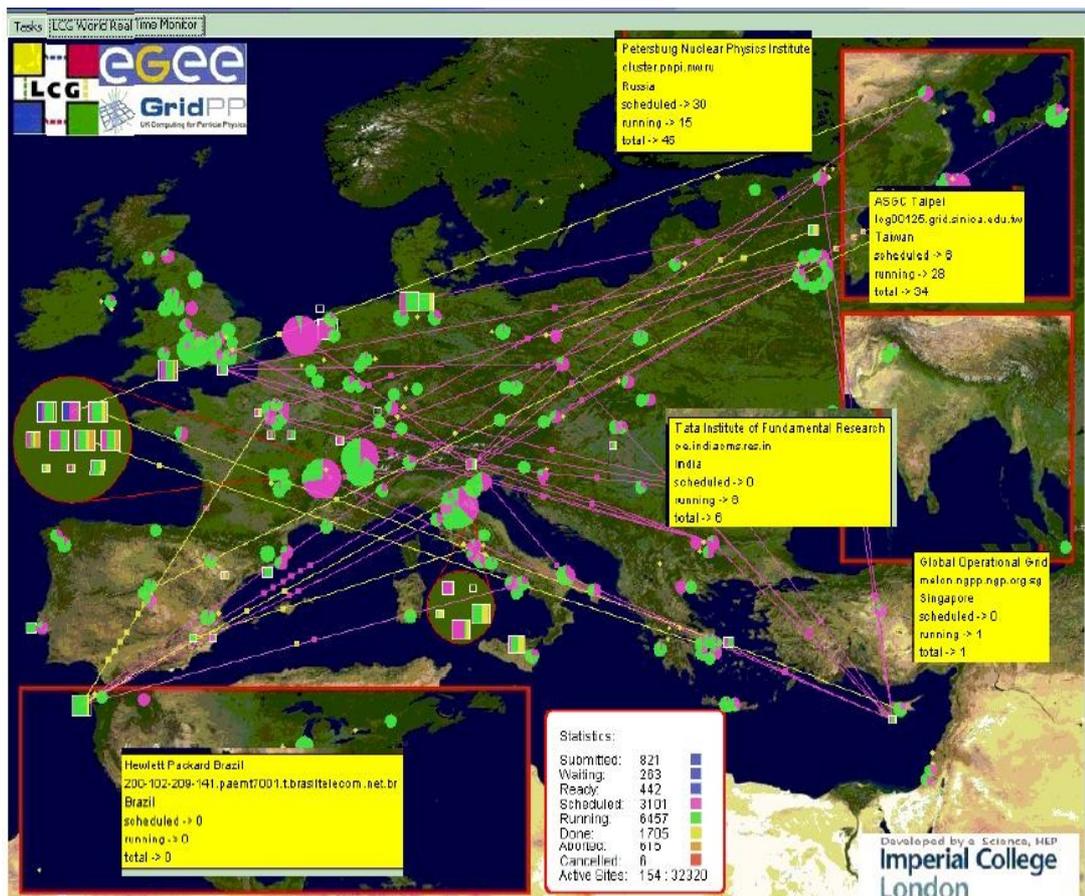


Рис. 12 Мгновенный снимок карты постоянного мониторинга, на которой показано распределение заданий по ресурсам в данный момент.

Помимо ресурсных центров и базовых грид-служб, очевидной необходимой предпосылкой для создания глобальной грид-среды является наличие глобальной высокоскоростных открытых сетей.

3.3.1 Базовая компьютерная сеть: GEANT

Сетевая технология уже достигла уровня, приемлемого для создания глобального грида, и номинальная пропускная способность сетей продолжает расти с достаточной скоростью. А скорость, с которой расширяется область охвата высокоскоростными сетями, - больше вопрос национальных бюджетов, чем технологическая проблема.

В проекте EGEE основой грид-среды стала исследовательская сеть GEANT [14]. Проект GEANT основан Консорциумом 27 европейских национальных исследовательских и образовательных сетей (NRENs) с целью усовершенствования предыдущего поколения пан-европейской исследовательской сети TEN-155 путем создания магистральной основы, работающей на скоростях, превышающих 1 Гбит/сек. Эта сеть дополняет и соединяет национальные научные и образовательные сети в разных странах Европы. Она объединяет более 3 тысяч научно-исследовательских и учебных заведений, 3 миллиона индивидуальных пользователей из 35 стран Европы. Благодаря большой пропускной способности и высокому уровню доступа сеть GEANT дает пользователям возможность применять новые прикладные программы с целью обеспечения их исследований в различных областях науки.

3.3.2 Инфраструктура безопасной работы в грид-среде.

Ключевую роль в гриде играют вопросы безопасности. Это связано с тем, что сама идея грида заключена в предоставлении пользователям возможности удаленно исполнять программы. В связи с этим, нужно, с одной стороны, обеспечить удобный и единообразный авторизованный доступ к ресурсам, учет их использования, а с другой стороны, защитить ресурсы и данные от несанкционированного использования. Как подробно объяснено в разделе 2.5, для достижения поставленной задачи в качестве основы системы безопасности в глобальных грид-системах и, в частности в EGEE, используется технология несимметричных ключей и стандарт X509. Это позволило заменить слабый в отношении безопасного доступа метод авторизации при помощи имени входа и паспорта на мощную и надежную технологию основанную на сертификатах. Важно заметить, что эта технология позволяет перейти от модели доверительных отношений со многими тысячами пользователей к отношению доверия с несколькими центрами сертификации.

В России первый центр выдачи грид-сертификатов был организован в НИИЯФ МГУ в 2004 году. В настоящее время в соответствии с распределением работ в рамках проекта EGEE, российский сертификационный центр переведен в РИЦ «Курчатовский институт».

3.3.3 Основная организационная единица для пользователей грида – виртуальная организация

Как уже неоднократно упоминалось, одним из центральных понятий грида является понятие виртуальной организации. Виртуальная организация (ВО) является динамическим сообществом людей, которые совместно используют грид-ресурсы в соответствии с согласованными между ними и собственниками ресурсных центров правилами. Эти правила регулируют доступ ко всем типам средств, включая компьютеры, программное обеспечение и данные. Непосредственное управление виртуальными организациями осуществляется администратором (менеджером) ВО, ответственным за ведение записей о статусе членов ВО в базе данных ВО, то есть осуществляющим (после соответствующих проверок) включение (или исключение) пользователей в число членов ВО, устанавливающее их полномочия и обновляющее информацию о пользователях. Техническая поддержка базы данных о составе и структуре ВО осуществляется на основе сервиса управления виртуальными организациями VOMS (п. 2.5.4), который обеспечивает детальную авторизацию пользователей грид-инфраструктуры.

Наличие ВО – существенное организационное отличие грида от Всемирной паутины (WWW), где пользователи работают индивидуально. Обычно в виртуальную организацию объединяются научные сотрудники из различных институтов, городов и стран. ВО предоставляет возможность образовывать временные межинститутские и межнациональные объединения исследователей (коллаборации) для решения крупных актуальных проблем. При этом благодаря грид-технологиям они получают доступ к объединенным ресурсам, работать с единым программным обеспечением, что обеспечивает достоверность и сравнимость результатов, полученных разными группами исследователей.

В настоящее время в EGEE существуют несколько десятков ВО, из таких областей как биомедицина, физика высоких энергий, физика термоядерного синтеза, астрофизика, науки о Земле, материаловедение, мультимедийные технологии, моделирование процессов на финансовых рынках и других. Более подробно о прикладных задачах, решаемых с помощью грида EGEE, мы расскажем в последнем разделе.

3.4 RDIG: российские ученые в проекте EGEE

Чтобы обеспечить полномасштабное участие России в осуществлении проектов EGEE/LCG в сентябре 2003 года был образован Российский консорциум РДИГ

(Российский ГРИД для интенсивных операций с данными - Russian Data Intensive GRID, RDIG) [16]. Меморандум о создании консорциума был подписан руководителями восьми крупных институтов: Института физики высоких энергий (Протвино), Института математических проблем биологии (Пушино), Института теоретической и экспериментальной физики (Москва), Объединенного института ядерных исследований (Дубна), Института прикладной математики им. М.В. Келдыша (Москва), НИИ Ядерной физики МГУ (Москва), Петербургского института ядерной физики (Санкт-Петербург) и РНЦ "Курчатовский институт" (Москва).

Консорциум РДИГ входит в структуру EGEE в качестве региональной федерации "Россия" и его целью является создание действующей грид-инфраструктуры в России, что включает в себя:

- наращивание вычислительных ресурсов и ресурсов хранения данных российского сегмента грид-среды;
- обеспечение надежной и быстрой сетевой инфраструктуры, в том числе с выходом на европейских партнеров проекта;
- обеспечение работы базовых грид-сервисов в российском сегменте;
- создание Регионального операционного центра (РОЦ; Regional Operations Center - ROC);
- помощь ресурсным центрам в установке ППО и поддержка его функционирования;
- поддержка пользователей грида;
- участие в предоставлении ресурсов для исследований в важных прикладных областях (в частности, в области биомедицины, термоядерного синтеза, физики высоких энергий и космофизики);
- управление функционированием инфраструктуры: регистрация пользователей, региональных виртуальных организаций и мониторинг;
- популяризация грид-технологий и вовлечение новых пользователей из научных и производственных кругов.

Основной набор базовых грид-служб RDIG сосредоточен в НИИЯФ МГУ, но часть служб обеспечиваются и другими организациями в соответствии с распределением ответственности между участниками проекта EGEE. Благодаря наличию полного набора базовых служб, RDIG способен работать как независимо от остальной инфраструктуры EGEE, так и в качестве сегмента глобальной грид-инфраструктуры проекта EGEE.

В настоящее время в рамках RDIG работают более 100 ученых и специалистов в области компьютерных технологий, существует более десятка ресурсных центров, предоставляющих ресурсы для грид-среды, с общим числом процессоров более 500 и с общим объемом хранилищ данных около 300 Терабайт.

Например, ресурсный центр НИИЯФ МГУ включает в себя, в настоящее время, кластер на основе рабочих станций Intel-Хеон рабочей частоты 3ГГц и объемом памяти 1-2 Гб на рабочий узел (процессор или ядро - в случае процессоров с несколькими ядрами). Всего доступно через грид более 100 рабочих узлов. Ресурсы хранения данных включают 3 дисковых файл-сервера общим объемом 9 Тб и ленточная роботизированная библиотека на 6 Тб. В ближайшем будущем планируется увеличить объем дисковой памяти до 21 Тб. Локальная сеть построена по технологии Гигабитной сети Ethernet. Такая же технология используется для выхода в Интернет. По мощности предоставляемых ресурсов, это один из крупнейших ресурсных грид-центров в России.

3.4.1 Региональные российские виртуальные организации

Цель создания региональных ВО, функционирующих только в рамках РДИГ, - предоставление вычислительных ресурсов для национальных научных проектов и апробирование использования грид-технологий в новых прикладных областях, имея в виду их последующее включение во всю инфраструктуру EGEE.

В настоящее время в рамках РДИГ функционируют виртуальные организации в области космофизики (AMS), физики высоких энергий (PHOTON), термоядерного синтеза (Fusion_RDIG), геофизики (eEarth). Существуют также две тестовые ВО: RDTEAM – для мониторинга и тестирования грид-инфраструктуры, и RGStest - неспециализированная ВО для целей пробного использования грида в новых прикладных областях, другими словами, эта ВО предназначена для исследователей, которые пока не приняли решение о создании своей ВО, но хотели бы практически ознакомиться с преимуществами и особенностями работы в грид-инфраструктуре.

С текущей информацией о действующих российских региональных ВО можно ознакомиться на сайте российского Регионального операционного центра: <http://grid.sinp.msu.ru/grid/roc/voinrdig>.

3.5 Грид-среда EGEE глазами пользователей и провайдеров ресурсов

Прежде, чем перейти к заключительному разделу этой части, посвященному прикладным задачам, которые уже решаются с помощью грида EGEE, кратко опишем как эта инфраструктура выглядит не со стороны, а с точки зрения провайдера ресурсов и пользователей.

3.5.1 Грид-среда глазами провайдера ресурсов

Предположим, что какая-то организация, обладающая вычислительными ресурсами и ресурсами хранения данных, решила предоставить их для совместного использования в рамках грид-инфраструктуры EGEE. Побудительные мотивы для такого решения, вообще говоря, могут быть разными, но, как правило, это надежда (вполне оправданная), что предоставив свои ресурсы, будет, в свою очередь, получен доступ к значительно большим ресурсам всей инфраструктуры.

Первым шагом для достижения этой цели должно быть обращение в Российский операционный центр, а более конкретно – начать надо с изучения соответствующей инструкции на сайте Центра: <http://grid.sinp.msu.ru> (инструкцию можно найти по ссылке «Как присоединиться в качестве ресурсного центра»). Далее с организационной точки зрения надо пользоваться указаниями этой инструкции.

Кроме того, необходимо, конечно, изучить инструкции по установке соответствующих компонентов (вычислительного элемента и ресурса хранения данных) ППО gLite и инструкции пользователя на сайте [22].

После установки ППО и регистрации ресурсного центра в инфраструктуре EGEE (как это делается указано в инструкции операционного центра), необходимо решить какие виртуальные организации будут поддерживаться данным ресурсным центром, то есть члены каких ВО смогут выполнять задания на данных ресурсах. Это решение может зависеть от многих обстоятельств и в первую очередь – от целей предоставления ресурсов в грид-инфраструктуру. Далее политику предоставления ресурсов необходимо детализировать – решить какие ВО и/или группы пользователей внутри той или иной ВО будут пользоваться теми или иными преимуществами или, напротив, иметь ограниченный доступ к ресурсам. Технически это делается с помощью отображения групп пользователей на формальных локальных Unix/Linux-пользователей с теми или иными правами.

Дальнейшая деятельность по обеспечению работы ресурсного центра сводится к постоянному мониторингу его работы, устранению неполадок, корректировке политики предоставления ресурсов – в целом все это аналогично администрированию любого компьютерно-сетевого комплекса.

3.5.2 Грид-среда глазами пользователя

Начальные действия будущего пользователя грида существенно зависят от того, в каком качестве он хочет присоединиться к грид-сообществу: в качестве члена уже существующей виртуальной организации или создателя новой ВО. Но в любом случае – первый шаг, это посещение веб-сайта операционного центра и изучение соответствующих инструкций (сайт <http://grid.sinp.msu.ru> и далее - по соответствующей ссылке «Как присоединиться в качестве...»).

Естественно, что организационно и технически присоединиться к уже существующей ВО проще. Основные шаги – это получение электронного сертификата и регистрация (с помощью веб-интерфейса службы VOMS, п. 2.5.4) в выбранной ВО. Последнее может потребовать дополнительных переговоров – в процессе обработки заявки на включение - с менеджментом ВО (координаты – на веб-странице операционного центра <http://grid.sinp.msu.ru/grid/roc/voinrdig>) о возможности и условиях включения данного заявителя в ВО. После успешной регистрации в какой-либо ВО, пользователь может использовать грид-ресурсы поддерживающие данную ВО. Конечно, для этого необходимо изучить инструкцию пользователя или хотя бы часть, касающуюся запуска и описания заданий, а также использования распределенных каталогов данных.

Как видно, пользователь должен затратить определенные усилия и время, чтобы начать использовать грид-ресурсы. Поэтому важно заранее оценить насколько эффективным будет грид для решения конкретной прикладной задачи. Как мы уже подчеркивали (часть I), грид-инфраструктура в первую очередь эффективна для решения задач, которые разбиваются на большое число независимых заданий и требуют обработки большого объема данных. Но даже, если задача пользователя относится к такому классу, вряд ли использование грида будет эффективным, если этот большой набор независимых заданий запускается вручную. Поэтому пользователь должен быть готов к тому, что потребуются создать специализированную систему запуска его заданий через интерфейс пользователя для достижения приемлемой эффективности. Кроме того, необходимо отчетливо понимать, что в настоящее время все рабочие узлы инфраструктуры EGEE (где собственно и выполняются задания) работают под управлением операционной системы (ОС) Linux (более того, конкретного дистрибутива – Scientific Linux). Поэтому задания должны быть подготовлены для выполнения в среде этой ОС. Обнадеживающим фактом для расширения класса выполняемых задач является впечатляющий прогресс в технологии виртуализации вычислительных ресурсов. Это позволяет надеяться, что в недалеком будущем появиться возможность по запросу конкретного задания разворачивать на рабочем узле виртуальный компьютер с подходящей этому заданию вычислительной средой. В частности, работа в этом направлении ведется в рамках проекта Globus (<http://workspace.globus.org>).

В случае создания новой виртуальной организации первым этапом также является ее регистрация в инфраструктуре EGEE/RDIG в соответствии с инструкцией на сайте операционного центра. Но затем наступает самый ответственный этап – переговоры с ресурсными центрами об их поддержке этой новой ВО. Поскольку денежных расчетов (по крайней мере в настоящее время) в инфраструктуре EGEE/RDIG не существует, необходимо найти какие-то убедительные аргументы для такой поддержки. Как правило, такие аргументы связаны со взаимным предоставлением грид-ресурсов. В случае успешного начала работы ВО, дальнейшая деятельность менеджера ВО связана с управлением составом организации (прием новых членов, исключение, распределение

членов ВО по группам и присваивание отдельным пользователям специальных полномочий), а также с взаимодействием с операционным центром и администраторами ресурсных центров при возникновении нештатных ситуаций.

3.6 Прикладные задачи в проекте EGEE

Как уже упоминалось, в настоящее время исследователи из целого ряда научных и производственных областей используют грид-инфраструктуру EGEE для решения своих задач. В их числе биомедицина, физика высоких энергий, физика термоядерного синтеза, астрофизика, науки о Земле, материаловедение, мультимедийные технологии, моделирование процессов на финансовых рынках и другие.

Очень важно, что многие приложения перешли от этапа тестирования процесса запуска своих задач в грид-среду к практической рутинной работе для получения новых результатов в своих областях. При этом эффективность загрузки грид-ресурсов EGEE достигает ~80-90%. Для иллюстрации прогресса в использовании грида на рис. 13 приведен график числа заданий запускаемых ежедневно в период с начала 2005 до середину 2006 года.

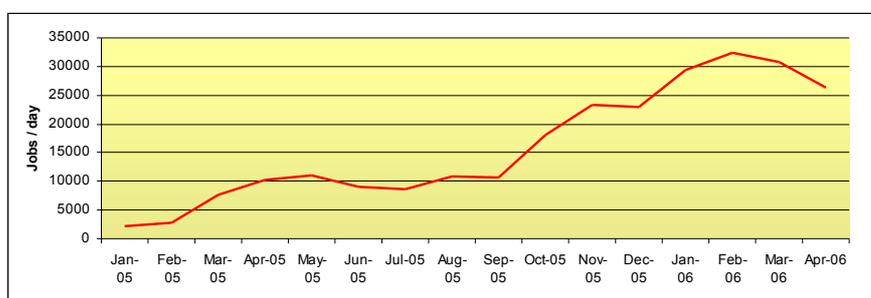


Рис.13 Интенсивность запуска заданий в грид-инфраструктуре EGEE в 2005-2006 годах

В этом разделе мы кратко опишем ряд прикладных задач и проектов, которые используют грид EGEE.

3.6.1 Физика высоких энергий: проект LCG

Главной целью проекта LCG (Large Hadron Collider Computing Grid) [12] является использование грид-среды для моделирования и обработки экспериментальных данных с Большого адронного коллайдера (Large Hadron Collider, LHC) [15], строящегося в Европейском центре ядерных исследований (ЦЕРН, CERN) в окрестностях Женевы (Швейцария). LHC – ускоритель протонов для исследования фундаментальных свойств субатомных частиц; начало его эксплуатации запланировано на 2008 год. Ожидаемый поток экспериментальных данных, который необходимо будет обрабатывать после начала работы ускорителя, будет чрезвычайно большим – около 15 Петабайт в год. Сосредоточить компьютерные ресурсы, необходимые для обработки такого потока, непосредственно в ЦЕРН является совершенно трудно выполнимой задачей – и по техническим, и по финансовым причинам.

Мы начали рассказ о прикладных областях с физики высоких энергий потому, что исторически именно ЦЕРН с институтами-партнерами из разных стран мира, участвующими в проекте LHC, инициировал – в рамках проекта LCG - создание сначала пан-европейской, а затем глобальной грид-системы для решения задач проекта LHC, которая затем переросла в универсальную грид-инфраструктуру EGEE.

Грид-система, разработанная в LCG, проходит пробную эксплуатацию в рамках подготовки к пуску LHC. В настоящее время она используется для моделирования потоков данных четырех больших экспериментов (ALICE, ATLAS, CMS и LHCb), которые

будут осуществлены на коллайдере в рабочем режиме ускорителя. Очень кратко охарактеризуем эти эксперименты:

- ALICE (A Large Ion Collider Experiment) – эксперимент по изучению физики сильных взаимодействий при сверхвысоких плотностях, где ожидается образование нового состояния вещества - кварк-глюонной плазмы.
- В эксперименте ATLAS (A Toroidal LHC ApparatuS) будут изучаться глубинные основы строения вещества и фундаментальные силы, сформировавшие Вселенную.
- CMS (Compact Muon Solenoid) – детектор для новых исследований, в ходе которых будет предпринята попытка подтвердить или опровергнуть единую теорию фундаментальных сил природы.
- LHCb (Large Hadron Collider beauty) – эксперимент по изучению нарушения симметрии заряда и чётности. Этот эффект может являться причиной неравновесия между веществом и антивеществом при рождении Вселенной.

В 2001-2006 годах коллективами всех этих экспериментов были осуществлены сеансы по массовой генерации потоков модельных данных. Были протестированы основные типы вычислительных задач (моделирование, реконструкция событий в ускорителе и их анализ) на мощностях, сравнимых по масштабу с рабочими параметрами LHC. Результатами этих тестов стали рекордные скорости передачи и распределенной обработки данных, а также ранее недоступные возможности хранения данных.

Сообщество исследователей в области физики высоких энергий стало первым в EGEE, где началась эксплуатация приложений, и до сих пор является крупнейшим пользователем Грид-инфраструктуры EGEE. Помимо экспериментов на LHC, грид EGEE используют и другие эксперименты в этой области исследований - BaBar, CDF, H1, ZEUS и DØ.

Поскольку по своей природе приложения для физики высоких энергий предъявляют очень высокие требования к инфраструктуре EGEE, они очень сильно способствуют улучшению сервисов EGEE и выработке принципиальных подходов к их развитию. Это относится ко всем сервисам – от документации и поддержки пользователей до разработки промежуточного программного обеспечения. Кроме того, в ходе экспериментов в физике высоких энергий создаются ценные компоненты ППО, которые можно считать прототипами для всего сообщества пользователей грид-технологий.

Российские ученые активно участвуют как в подготовке физических экспериментов, так и развитии грид-среды для обработки данных, которые либо уже получены (для действующих экспериментов), либо будут получены в близком будущем (как в случае LHC).

3.6.2 Астрофизические приложения

Два приложения в области астрофизики, которые поддерживаются EGEE, связаны с обработкой данных со спутника Planck и с телескопа MAGIC.

Спутник Европейского космического агентства (European Space Agency - ESA) Planck будет запущен в 2008 году и предназначен для создания «микроволнового атласа» неба: детекторы, которые будут установлены на спутнике, по крайней мере дважды выполнят полное сканирование небесной сферы в микроволновом диапазоне (30 ÷ 850 GHz) с недоступной ранее полнотой, стабильностью, точностью и чувствительностью. Грид EGEE обеспечит предварительное моделирование процесса обработки данных со спутника, а в последствии и реальную их обработку (ожидается ~100МВ данных в день; полный объем данных составит около 100GB).

Принцип работы телескопа MAGIC, который работает с 2004 года, основан на так называемом эффекте Черенкова. Телескоп расположен на Канарских островах и

используется для исследования высокоэнергетических частиц, приходящих из космоса – космических лучей и вызываемых ими «ливней» заряженных частиц в высоких слоях атмосферы. Пробная обработка данных с этого телескопа на ресурсах EGEE (Data Challenge) началась в 2005 году. Важным этапом этого проекта будет ввод в строй второго аналогичного телескопа (он будет расположен в 85 км от MAGIC). Это позволит существенно улучшить возможности исследования космических лучей, но и существенно увеличит поток данных, которые будут обрабатываться в грид-инфраструктуре EGEE.

3.6.3 Приложения в области ядерного синтеза

ITER (the International Thermonuclear Experimental Reactor – "Международный экспериментальный термоядерный реактор") – это международный проект исследований и разработок, цель которого – показать научную и техническую возможность использования энергии термоядерного синтеза. Реактор будет построен в г. Кадараш (Cadarache) во Франции. Водородная плазма в нем будет удерживаться в форме тора при температуре выше миллиона градусов, что, в принципе, приблизительно к 2016 г. может позволить создать управляемый термоядерный реактор мощностью порядка 500 мегаватт.

Управляющий комитет (Steering Committee), действующий в рамках Европейского соглашения по разработкам в области термоядерного синтеза (European Fusion Development Agreement (EFDA)), учредил группу, задача которой – изучить перспективу потребностей в вычислительных ресурсах у европейского сообщества исследователей в области ядерного синтеза. Уже показана способность грид-инфраструктуры удовлетворять эти потребности. В настоящее время прикладные задачи EFDA обрабатываются в грид-инфраструктуре EGEE. В будущем не исключено создание специализированной грид-инфраструктуры проекта ITER (столь ресурсоемки задачи этого проекта).

Сейчас в грид-инфраструктуре EGEE работают следующие приложения для исследований ядерного синтеза:

- Massive Ray Tracing ("Прослеживание траектории лучей на основе большой статистики") рассчитывает траекторию микроволнового пучка в плазме.
- Kinetic Transport ("Кинетический перенос") рассчитывает явления кинетического переноса посредством отслеживания орбит большого числа независимых частиц, которые претерпевают столкновения с фоновой плазмой, характеризующейся температурой, плотностью и электрическим полем. Окончательные траектории используются для расчёта важных свойств явлений переноса в плазме – потока частиц, тепла и других.
- Stellarator – приложение для оптимизации стелларатора – установки постоянного режима для магнитного удержания плазмы, где происходит термоядерный синтез и нет токов.

Кроме того, как уже было отмечено в п. 3.4.1, отдельную региональную виртуальную организацию основали российские исследователи – в рамках RDIG. Сейчас эта организация разрабатывает свои собственные методики применения грид-технологий, формулирует требования к промежуточному программному обеспечению и размещает в грид-инфраструктуре свои первые приложения.

3.6.4 Решение задач биомедицины с помощью грид-среды EGEE

Биомедицина – одна из основных прикладных областей, которые используют грид EGEE. Задачи в этой области подразделяются на три направления:

- обработка изображений в медицине;
- биоинформатика;

- поиск новых лекарств.

Для задач обработки медицинских изображений грид оказался настолько удачным решением, что с EGEE сотрудничает целый ряд проектов этого направления. Кратко перечислим некоторые из них.

- Программный пакет GATE осуществляет компьютерное моделирование радиотерапии, что позволяет планировать его применение на основе графических результатов обследования пациента. Грид-инфраструктура EGEE используется для существенного уменьшения времени осуществления компьютерного моделирования для конкретного пациента – так, чтобы оно было допустимым с точки зрения использования в клиниках.
- Клинические системы помощи при принятии решений (Clinical Decision Support System - CDSS) используют классификацию изображений для того, чтобы помочь врачам при принятии решений о выборе методов лечения. Грид используется для сбора больших объемов данных и для эффективного обучения классифицирующего ПО на этих больших наборах данных.
- Приложение Pharmacokinetics изучает распространение контрастного вещества в печени, используя последовательность изображений, полученных методом магнитного резонанса. Артефакты, связанные с движениями пациента, делают невозможным непосредственное сравнение изображений. Однако специальная корректирующая обработка изображений (на основе нахождения подобных фрагментов) позволяет анализировать такую последовательность изображений. Грид-инфраструктура EGEE позволяет делать это за разумное время.
- Похожую задачу решает с помощью грида ПО SiMRI3D - анализ изображений, получаемых методом магнитного резонанса (MP) от детально известных источников, изучения артефактов и дальнейшего развития метода наборов MP-изображений.
- gPTM3D – интерактивное восстановление трехмерных медицинских изображений: например, изображения всего объема сложных органов. Поскольку это приложение работает в интерактивном режиме так, что время получения результатов выполнения заданий должно быть строго ограниченным, некоторые ресурсные центры грид-инфраструктуры EGEE обрабатывают задания этого приложения с высоким приоритетом.
- Программный пакет SPM используется для ранней диагностики болезни Альцгеймера. Этот метод основан на сравнении показателей пациента с подозрением на заболевание к большому набору нормальных случаев. Грид-технологии обеспечивают доступ к распределенным данным и позволяют быстро проводить этот сравнительный анализ.
- Бронзовый Стандарт (Bronze Standard) это исследовательский проект, целью которого является оценка алгоритмов получения медицинских изображений. Объем данных, которыми при этом приходится манипулировать, превышает возможности одиночных стандартных компьютеров, но приложение может легко быть распределено по ресурсам грида и выполнено за разумное время.

Направление биоинформатики занимается анализом последовательностей генов. В сферу его интересов входят геномика, протеомика и филогения.

- Проект GPS@ (Grid Protein Sequence Analysis – "Анализ белковых цепочек на основе грид-технологий") поддерживает веб-портал, предоставляющий удобный интерфейс доступа к соответствующим биоинформационным ресурсам в грид-инфраструктуре EGEE. Доступен также прототип этого портала, где есть интерфейс с 13 программами в грид-инфраструктуре из 46 программ оригинального портала.

- Пакет `xmipp_MLrefine` обеспечивает трехмерный структурный анализ больших макромолекулярных комплексов. В процессе восстановления структуры комплексов совместно используется множество разных изображений исследуемого образца. Эти изображения зачастую оказываются искаженными и для составления наиболее соответствующей экспериментальным данным модели необходимо делать много итераций, что требует значительных грид-ресурсов.
- Похожую задачу решает программный пакет `Xmipp_assign_multiple_CTFs` для изображений, получаемых на электронном микроскопе, - они страдают разными формами аберрации. Различие между теоретическим и экспериментальным изображением-проекцией математически описывается функцией контрастного переноса. Пакет `Xmipp_assign_multiple_CTFs` выполняет моделирование для нахождения этой функции.
- Задачей проекта SPLATCHE (SPatial And Temporal Coalescences in Heterogeneous Environment – "Пространственно-временные интеграции в разнородной окружающей среде") является моделирование эволюции генома человека. Оно позволяет восстановить расселение человека по Земле в географически правдоподобных ландшафтах и моделировать молекулярное разнообразие разных человеческих популяций.

Использование грида для разработки лекарств позволяет ускорить поиск новых лекарств посредством компьютерного моделирования структуры и динамики белков. Дадим краткое описание некоторых проектов этого направления.

- Drug Discovery занимается анализом состыковки молекул (докинга), целью которого является выяснение того, насколько эффективно конкретные лекарства присоединяются к определенным участкам вируса-мишени. Успешными оказались приложения для поиска средств от малярии и птичьего гриппа; планируется поиск лекарств и от других вирусов.
- GridGRAMM предоставляет веб-интерфейс для расчетов докинга молекул. Расчеты включают оценку качества докинга и разные методы анализа трехмерной структуры комплекса. Молекулярный докинг может применяться для изучения межмолекулярных взаимодействий, изучения взаимодействий между ферментами и субстратом, разработки лекарств и понимания патологических мутаций.
- GROCK (Grid Dock) веб-реализация инструментария для отбора возможных межмолекулярных взаимодействий из огромного объема информации. Пользователи могут исследовать одну молекулу относительно целой базы данных по известным структурам.

3.6.4.1 Drug Discovery: борьба с массовыми болезнями с помощью грида

Расскажем несколько более подробно о приложении Drug Discovery (поиск лекарств), работающем в рамках EGEE. Он предназначен для поиска принципиально новых лекарств от массовых заболеваний, например от малярии – болезни, от которой ежегодно страдают 300 миллионов человек, а миллион умирает. Причем ситуация ухудшается из-за повышения устойчивости болезни к существующим лечебным препаратам.

Это приложение было инициировано и реализовано Институтом алгоритмов и научных вычислений им. Фраунгофера SCAI (Fraunhofer Institute for Algorithms and Scientific Computing) в Германии и Лабораторией корпускулярной физики в Клермон-Ферране, Франция (IN2P3). Оно позволяет повысить возможности доступа фармацевтических

компаний и академических исследовательских институтов к разнообразной, сложной и распределенной информации о болезнях и обеспечить возможность совместных исследований по поиску новых лекарств.

Приложение основано на возможности вычислять вероятность того, что новое потенциальное лекарство войдет в прямой контакт с активной частью одного из паразитных белков малярии. Обычно такие вычисления выполняются на кластерах ПК и ограничиваются примерно 100 000 кандидатами на новое лекарство. В сеансе массовой обработки данных, названном WISDOM (Wide In Silico Docking On Malaria) и проведенном в грид-среде EGEE в августе 2005 года, исследовано свыше 46 миллионов кандидатов. В сеансе одновременно использовалось 1000 вычислительных машин в 15 странах по всему миру; на одном персональном компьютере для выполнения такой работы потребовалось бы 80 лет. Успех сеанса WISDOM продемонстрировал, какую помощь может оказать грид в исследованиях по поиску лекарств, значительно ускоряя весь процесс разработки.

А в течение апреля 2006 года лаборатории Азии и Европы, используя помимо EGEE грид-инфраструктуры AuverGrid и TWGrid, провели совместную работу по анализу 300000 возможных компонент лекарства против вируса птичьего гриппа H5N1. Целью работы был поиск возможных соединений, которые могли бы подавлять активность фермента на поверхности вируса гриппа, так называемой нейраминидазы, подтип N1. Чтобы свести 300 000 компонент к 8 различным целевым структурам нейраминидаз гриппа А, были задействованы 2000 компьютеров, работавших в течение 4 недель в апреле. Если бы работал один компьютер, ему понадобилось бы 100 лет для решения этой задачи. Было создано и сохранено в реляционной базе данных более 60 000 выходных файлов с объемом информации 600 Гб. В настоящее время потенциальные компоненты лекарства против птичьего гриппа идентифицируются и классифицируются.

Следующие шаги в развитии приложения по поиску новых лекарств включают классификацию большого количества данных, чтобы идентифицировать потенциальные лекарства, используемые в лечении ряда заболеваний, и сократить разрыв между такими «виртуальными кандидатами» и традиционной разработкой лекарств. Это может привести к появлению значительного числа физических молекул - кандидатов для лекарств, которые в дальнейшем могут быть доведены до уровня реальных терапевтических компонентов.

3.6.5 Приложения в области вычислительной химии

В грид-инфраструктуре размещены и работают в постоянном режиме несколько приложений в области вычислительной химии:

- Приложение GEMS (Grid Enabled Molecular Simulator – "Молекулярный симулятор на основе грид-технологий") используется для моделирования динамики реакций в сложных химических системах. Для его эксплуатации в инфраструктуре EGEE создана виртуальная организация CompChem.
- AVCTraj рассчитывает наблюдаемые величины атом-диатомного взаимодействия в газообразной фазе. События генерируются методом Монте-Карло. Программа подключена к среде молекулярной виртуальной реальности, где на мониторы выводятся результаты моделирования.
- Venus рассчитывает сечения и коэффициенты скорости элементарных химических реакций, моделируя столкновения атомов и молекул при начальных условиях, полученных методом Монте-Карло.
- Приложение DI-Poly моделирует молекулярную динамику сложных систем. Оно стало стандартом де-факто в сообществах вычислительной химии и

вычислительной биологии.

- Приложение RWAVEP выполняет квантовомеханические расчёты вероятности разных химических реакций, применяя подход, основанный на понятии волнового пакета. Для разных наборов исходных состояний генерируются разные события.

В ближайшем будущем виртуальная организация CompChem развернёт новые приложения, например:

- COLUMBUS – комплект программ для сложных расчётов ab initio электронной структуры молекул. Программы предназначены, главным образом, для расширенных многоссылочных расчётов основного и возбуждённых состояний электронной оболочки атомов и молекул.
- GAMESS – программа для выполнения ab initio расчётов в области квантовой молекулярной химии, позволяющая вычислять волновые функции SCF. Возможны следующие корреляционные поправки к этим волновым функциям: учёт зависимости взаимодействия от конфигурации; поправки на основе теории возмущений второго порядка; поправки на основе представлений о спаренных кластерах; возможно также приближение в рамках функциональной теории плотности.

Кроме того, виртуальная организация CompChem будет экспериментировать с системой CHARON, чтобы создать настраиваемый пользователем интерфейс с гридом, отвечающий специфическим требованиям сообщества вычислительной химии.

Важно отметить, что большинство задач в этой области, в том числе и среди приложений, перечисленных выше, являются истинно параллельными (по терминологии части I, то есть требуют межпроцессорных обменов) и могут выполняться только целиком на кластерах с соответствующей сетевой обвязкой и вычислительной средой (при наличии библиотеки MPI).

3.6.6 Приложения для наук о Земле и геофизики

Сообществами представителей наук о Земле и геофизики в EGEE поддерживаются пять специализированных приложений в области гидрологии, наблюдения за поверхностью земли, климатологии и физики твердой Земли. Созданы две виртуальные организации: ESR (Earth Science Research – "Исследования в области наук о Земле") - для академических учреждений и EGEODE (Expanding GEOsciences on Demand – "Развитие наук о Земле по запросам"), основанную во Франции частной компанией CGG (Compagnie Générale de Géophysique).

В рамках ESR развернуто несколько исследовательских приложений:

- Анализ наблюдений за поверхностью земли: грид-инфраструктура помогает вести анализ характеристик озонового слоя, измеряемых спутником GOME, а также использует данные спутникового эксперимента ERS/SAR по обнаружению утечек нефти.
- Приложения в области физики суши сосредоточены на механизмах землетрясений и численном моделировании землетрясений в сложных трехмерных геологических моделях.
- В области гидрологии изучается проникновение морской воды в прибрежный водоносный слой в бассейне Средиземного моря, обусловленное хозяйственной деятельностью человека. Работа приложения представляет собой моделирование методом Монте Карло, основанное на трехмерной модели движения грунтовых вод и переноса соли с учетом плотности среды.

- В области климатологии выполняется каскад процессов моделирования, исходными данными для которых являются экспериментальные данные.

Виртуальная организация EGEODE функционирует для первого производственного приложения EGEE – программный пакет Geocluster используется для обработки сейсмических данных и исследования состава слоев земной коры. Необходимо отметить, что задаче привлечения производственных приложений в рамках проекта EGEE уделяется большое внимание. Достижению этой цели способствуют Производственный форум EGEE (Industry Forum), Группа производственных задач (Industry Task Force) и программа EGEE по работе с бизнес-партнерами (EGEE Buisness Associate), которые должны сделать открытой для производства инфраструктуру и ноу-хау грида EGEE.

3.6.6.1 Грид для предсказания извержений вулканов

Один из способов предсказания извержения вулканов основан на преобразовании геофизической информации о поведении вулканов в звуковые волны. Затем картина звуковых волн – своеобразная «музыка» вулканов – анализируется, и это позволяет сделать предсказания поведения вулканов в близком будущем, в том числе предсказать извержения. В сотрудничестве с проектом EELA (европейско-латиноамериканская инфраструктура), EGEE предоставляет сетевые и вычислительные ресурсы для такого анализа поведения вулканов Этна (на Сицилии) и Тунгурахау (Эквадор).

3.6.7 Финансы и мультимедиа

К концу 2006 года эти две области приложений в инфраструктуре EGEE были одними из самых новых и только начинали свою деятельность.

Мультимедийное приложение тестировалось на системах отладки в виртуальной грид-лаборатории GILDA [18] проекта EGEE.

Работа над финансовыми приложениями включает сотрудничество с Международным центром теоретической физики им. Абдуса Салама, который разворачивает итальянскую национальную грид-инфраструктуру для финансово-экономических исследований в рамках проекта Egrid, финансируемого министерством образования и науки Италии.

3.6.8 Грид-приложение в области археологии

Задачей приложения Археогрид (ArchaeoGRID) – с помощью грид-инфраструктуры EGEE и распределенных систем ряда других родственных проектов (в частности, проекта создания распределенных электронных хранилищ данных DILIGENT) осуществлять комплексный анализ сложных и разнородных археологических данных, полученных при раскопках и полевых исследованиях. Эти данные могут относиться к физике, химии, науке о Земле, биологии, географии, антропологии, социологии. С развитием археологических методов и теорий развиваются и способы получения данных. Кроме того, археологические данные теряют часть своего значения, когда рассматриваются вне начального пространственно-временного контекста, в котором они были получены. Этот пространственный и временной контекст современных археологических исследований простирается от места раскопок до гораздо больших регионов и охватывает длительный период времени. Заключительная стадия археологической работы также очень сложна, она включает визуализацию результатов (цифровые карты, виртуальные окружающие среды, двумерные и трехмерные изображения и так далее) и текстовое описание, основанное на анализе результатов и теоретических знаниях.

Структура Археогрида должна объединить эти различные подходы и данные в единую систему, удобную для анализа (который также планируется проводить с использованием грид-ресурсов).

4 Заключение

Грид-технологии в целом и грид-инфраструктура EGEE в частности вступают в пору зрелости – происходит переход от тестовых испытаний и пробного обслуживания пилотных приложений к постоянной устойчивой работе по обслуживанию самых разнообразных прикладных областей науки и производства. В связи с этим перед разработчиками нового прикладного ПО, перед разработчиками грид-ПО, и перед персоналом, обеспечивающим функционирование грид-инфраструктуры EGEE встают новые масштабные задачи.

Масштаб внедрения грид-технологий в значительной степени будет зависеть от способности новых и существующих приложений быть развернутыми в этой среде. Поэтому тенденции в области разработки и реализации приложений являются критическими для широкого распространения грида. К настоящему времени лишь относительно небольшое число приложений были написаны специально для грид-систем и лишь небольшое количество нынешних коммерческих приложений, представляющихся перспективными для исполнения в грид-системах, были действительно развернуты в гриде или были переведены на грид. Но по мере возрастания практического интереса к грид-технологиям и к связанным с ним моделям инфраструктур, потребуются и новые элементы планирования, модели разработки и средства для создания и реализации предназначенных для грида приложений.

Другим важным новым обстоятельством, которое появляется при переходе к полнофункциональному режиму работы глобальной грид-инфраструктуры является то, что основная масса пользователей не являются и не хотят быть экспертами в области грид-технологий. Поэтому важно обеспечить простые (графические, основанные на технологии веб-порталов) интерфейсы к системам запуска и мониторинга заданий и управления данными в гриде. Примером такого подхода является разработка компании GridwiseTech Веб-портала для запуска в грид-инфраструктуру EGEE громоздких вычислительных задач, связанных с компьютерным моделированием сложных электронных устройств. Зачастую разработчики приборов не являются ИТ-специалистами и поэтому работа в грид-инфраструктуре с помощью «интерфейса командной строки» является для них непростой задачей. Веб-портал обеспечивает простой графический доступ к вычислительным ресурсам EGEE и позволят запускать задачи без установки громоздкого программного обеспечения на компьютерах пользователя – достаточно иметь обычный веб-браузер. Благодаря этому разработчики таких устройств (особенно это важно для производства медицинских приборов) могут получить высококачественные результаты уже на этапе разработки и без построения реальных прототипов устройств. Необходимо отметить, что компьютерное моделирование электронных приборов весьма ресурсоемко: одна задача моделирования может занимать несколько месяцев на нескольких десятках процессоров. Разработка GridwiseTech позволяет запускать набор таких задач на множестве процессоров, входящих в грид-инфраструктуру.

Как видно из этого примера, создание удобных интерфейсов для конкретных приложений является весьма важным для того, чтобы сделать грид-инфраструктуру EGEE и в целом грид-технологии доступными широкому кругу пользователей в области промышленного производства и других бизнес-приложений.

Еще одним исключительно важным условием широкого использования грида является обеспечение возможности выполнения прикладного задания в грид-среде независимо от среды его разработки. Как уже упоминалось в п. 3.5.2, основные надежды на решение этой задачи связаны с технологией виртуализации ресурсов.

Что касается собственно проекта EGEE, то планируется, что его основной следующей задачей как раз и станет обеспечение устойчивой и полнофункциональной работы грид-инфраструктуры, повышение уровня обслуживания уже существующих и привлечение новых прикладных проектов. Общий план развития грид-инфраструктуры EGEE

представлен на рис. 14.

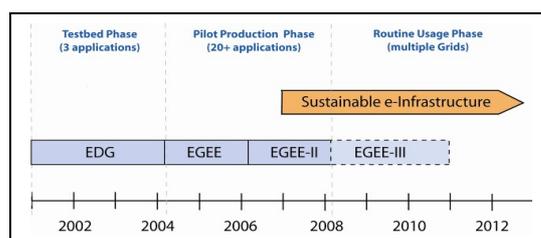


Рис.14 Общий план развития грид-инфраструктуры EGEE

Эта цель будет достигаться как в рамках EGEE, так и во взаимодействии с другими родственными проектами. Упомянем здесь только небольшую часть таких проектов, которые действуют уже сейчас. В проекте DILIGENT разрабатывается программное обеспечение для гридов, предназначенное для создания и поддержки цифровых библиотек. Проект DEGREE нацелен на распространение грид-технологий через большое и разнородное сообщество представителей наук о Земле. GRIDCC предполагает интегрировать в гриды контрольно-измерительные средства. Цель проекта BEinGrid – ускорить принятие грид-технологий посредством проведения нескольких бизнес-экспериментов и создания инструментария ППО.

С организационной точки зрения, планируется организовать грид EGEE как объединение национальных и региональных гридов с общим координирующим центром. Это показывает важность развития национальных грид-проектов – в частности Российского грида для интенсивных операций с данными (РДИГ).

В заключение еще раз подчеркнем, что грид-технологии позволяют решить проблемы, связанные с недостатком вычислительных мощностей для научных и производственных задач, требующих большого объема вычислений и/или обработки больших объемов данных. Это достигается объединением компьютерных ресурсов отдельных научных организаций, предприятий и компаний. Полученная в результате сеть компьютеров используется как единый ресурс.

Благодаря созданию действующей глобальной грид-системы проекта EGEE и уже проведенным с ее помощью вычислениям в различных областях стало очевидным, что грид станет новым важным инструментом, который позволит большому количеству ученых намного быстрее выполнять трудоемкие расчеты и обрабатывать большие объемы данных.

Работа выполнена при поддержке Европейской Комиссии (в рамках проекта EGEE), фонда РФФИ (грант № 05-07-90292) и Федерального агентства по науке и инновациям (грант NS1685.2003.2).

Краткий глоссарий терминов, связанных с веб/грид- технологиями¹

агент (agent)

Компонента программного обеспечения, которая функционирует от имени и по поручению пользователя, владельца ресурса или другого агента.

архитектура (architecture)

Формальное описание системы, определяющее ее цели, функции, внешне видимые свойства, и интерфейсы. Оно также включает описание внутренних компонентов системы и их отношений, наряду с принципами, управляющими ее дизайном, функционированием и возможной последующей эволюцией. Это описание включает программные компоненты, визуализированные свойства этих компонент, отношения между компонентами и ограничения на их использования. Программное обеспечение или система может состоять из многих уровней абстракции и многих фаз работы, и каждый уровень и каждая фаза могут иметь свою архитектуру.

асинхронный (asynchronous)

О взаимодействии говорится как об асинхронном, когда ассоциируемые сообщения хронологически и процедурно развязаны. Например, во взаимодействии "запрос - ответ" агент клиента может обрабатывать ответ в некоторый неопределенный момент в будущем, когда обнаружится, что этот ответ существует. К механизмам реализации такого взаимодействия относятся опрос, уведомление о получении другого сообщения и т.д.

авторизация (authorization)

Процесс определения по соответствующей информации для управления доступом, разрешены ли для субъекта указанные виды доступа к конкретному ресурсу. Обычно авторизация выполняется в контексте аутентификации. Если субъект аутентифицирован, он может быть авторизован для выполнения различных видов доступа.

аутентификация (authentication)

Процесс проверки и подтверждения, что потенциальный партнер по общению действительно представляет того, за кого он пытается себя выдать, с помощью некой уникальной информации (в простейшем случае — с помощью имени и пароля, в гриде чаще всего - с помощью электронных сертификатов).

виртуальная организация/ВО (virtual Organization)

Виртуальная организация (ВО) определяется как динамичное объединение пользователей, ресурсов и служб. ВО участвует в контрактах между поставщиками ресурсов и виртуальными организациями, которыми регулируются использование ресурсов и технические политики. Какая-нибудь группа пользователей и служб, входящих в состав ВО, может образовать в ее рамках группу, которая действует на основе договоренностей вышестоящей ВО.

вычислительный элемент (Computing element, CE)

¹ Этот краткий глоссарий основан на "Глоссарии терминов, принятых в OSG" и "Глоссарии Web-служб" [62], а также на материалах в Википедии [63].

В контексте грид-технологий термин "вычислительный элемент", используется для обозначения интерфейса ресурсного центра для запуска заданий на рабочие узлы.

делегирование (delegation)

Передача прав на осуществление тех или иных действий путем передачи запроса на выполнение работы или предложения ресурсов от пользователя или агента другому агенту. Принявшая поручение сторона наделяется четко определенным объемом ответственности и привилегий на каждом уровне передачи прав.

доступ к ресурсам

- управление доступом (access control)

Защита ресурсов от неавторизованного доступа; процесс, посредством которого использование ресурсов регулируется в соответствии с политикой владельцев ресурсов и разрешается только авторизованным пользователям и компонентам системы.

- права доступа (access rights)

Описание типов авторизованных взаимодействий, которые субъект может иметь с ресурсом (например, чтение, запись, исполнение, добавление, модификация, удаление).

имя файла,

- **логическое** (Logical File Name, LFN)

Глобальное уникальное имя файла в гриде, не зависящее от машины и места хранения.

- **физическое** (Physical file name, PFN)

URL физической реплики файла без учета протокола.

- **SURL** (Site Universal Resource Locator; Site URL)

определяет физическое местоположение файла или его реплики. В качестве SURL выступает полное имя SRM, понятное интерфейсу SRM элемента хранения данных (SE).

- **TURL** (Transport URL)

URL, который может использоваться, чтобы фактически передать файл, используя любой стандартный транспортный протокол. TURL начинается с протокола, используемого для передачи файла или прямого доступа к файлу через некоторый механизм ввода - вывода.

инфраструктура безопасности грида (Grid Security Infrastructure, GSI)

Компонента промежуточного программного обеспечения грида; основывается на понятии открытого ключа, входит в инструментальный пакет Globus.

кластер (cluster)

Доступная по сети группа рабочих узлов (при необходимости вместе с головным узлом), размещённая на некотором сайте. Другими словами, кластер это "контейнер", который группирует вместе компьютерные узлы или подкластеры.

компонента (component)

Объект программного обеспечения, предназначенный для взаимодействия с другими компонентами, инкапсулирующий некоторую функциональность или набор функциональностей. Компонента имеет четко определенный интерфейс и ведет себя заранее определенным образом, общим для всех компонент в рамках архитектуры.

менеджер ресурсов хранения (Storage Resource Manager, SRM)

Компонента промежуточного программного обеспечения грида для управления данными и виртуализации интерфейсов доступа к системам хранения. SRM не реализует передачу файлов непосредственно сам, а при необходимости обращается к службам передачи файлов, следит за работой этих служб и восстанавливает среду при возникновении аварийных ситуаций.

мониторинг/грид-мониторинг (monitoring/grid monitoring)

Грид-мониторинг подразумевает сбор, анализ и публикацию информации от распределенной инфраструктуры с целью определения статуса грид-ресурсов и хода выполнения заданий.

открытая архитектура грид-сервисов (Open Grid Services Architecture, OGSA)

OGSA является сервисно-ориентированной архитектурой грид-среды для использования в различных научных и промышленных областях. Эта архитектура опирается на веб-технологии, особенно на WSDL и SOAP, хотя в значительной степени предполагает независимость от способа обработки данных на транспортном уровне. Коротко говоря, OGSA является архитектурой распределенных информационно-вычислительных систем на основе сервисов, обеспечивающая интероперабельность в рамках неоднородной распределенной системы и возможность взаимодействия и совместной обработки информации различными типами ресурсов. По форме OGSA представляет из себя обобщение веб-сервисной архитектуры, отвечающее требованиям грид-компьютинга.

политика (policy)

Констатация четко определенных требований, условий или предпочтений, которые выставляются поставщиком и/или потребителем. Они используются в инфраструктуре при формулировании решений, определяющих те или иные действия и/или операции. В частности:

- **политика секретности (privacy policy)** - политика сбора, обработки, использования и раскрытия персональных данных другой стороны в результате взаимодействия;
- **политика безопасности (security policy)** - политика, определяющая как система или организация организуют службы безопасности для защиты ресурсов.

поставщик информации (Information Provider)

Программное обеспечение поставщика информации связывается с любой службой сбора данных, виртуально собирает данные некоторого типа и передает их информационной подсистеме грида.

приложение/прикладное программное обеспечение (application/application software)

Программы, предназначенные для выполнения определенных пользовательских задач и рассчитанные на непосредственное взаимодействие с пользователем. В грид-среде любое приложение при активизации (выполнении) содержит информацию, которая позволяет определить лицо, ответственное за выполнение этого приложения.

прокси (проху)

Агент, который ретранслирует сообщение агента-потребителя агенту-поставщику, выступая перед грид- или веб-службой в качестве потребителя.

промежуточное программное обеспечение/ППО (Middleware)

Слой программного обеспечения, состоящий из агентов, являющихся посредниками между различными компонентами крупного приложения. Зачастую ППО используется в распределённых приложениях, причём агентов, составляющих этот слой, может быть несколько.

протокол (protocol)

Набор формальных правил, описывающих, как пересылаются данные, особенно по сети. Протоколы низкого уровня определяют электрические и физические стандарты, которые должны соблюдаться, последовательность битов и байтов, передачу данных и обнаружение ошибок, коррекцию потока битов. Протоколы высокого уровня регулируют форматирование данных, включая синтаксис сообщений, терминал для диалога с компьютером, наборы символов, последовательность сообщений и т.д.

рабочий узел (Working Node, WN)

Отдельный хост кластера. Информация о вычислительном узле может быть видима гриду, но может быть и не видима - это зависит от способа администрирования кластера.

сайт (site)

Сайт – это используемое для администрирования логическое имя, обозначающее конкретный, стабильный, уникально идентифицируемый и тестируемый набор служб и ресурсов (вычислительных и ресурсов хранения данных).

сервис/служба (service)

Абстрактный ресурс, представляющий возможность выполнения задач, которые имеют четкие функции с точки зрения поставщиков и потребителей. Чтобы службой можно было воспользоваться, она должна быть реализована конкретным агентом поставщика.

В данном тексте термины "сервис" и "служба" используются как эквивалентные.

- **веб-сервис/служба (Web service)**

Веб-служба - это система программной поддержки взаимодействия "компьютер - компьютер" через сеть; веб-служба имеет интерфейс, описанный в формате, доступном для машинной обработки (WSDL); другие системы взаимодействуют с веб-службой так, как это определено ее описанием, с использованием SOAP-сообщений, передаваемых обычно по протоколу HTTP с XML-сериализацией в сочетании с другими веб-стандартами.

сервиса/службы интерфейс (service interface)

Определяет типы сообщений и шаблоны для обмена сообщениями, которые участвуют во взаимодействии со службой, а также условия, подразумеваемые этими сообщениями.

сервисно-ориентированная архитектура (Service Oriented Architecture, SOA)

Основа построения надежных распределенных систем, которые в качестве услуг предоставляют функциональные возможности, с дополнительным акцентом на

слабые связи между взаимодействующими сервисами. Базируется на наборе компонент, к которым можно обратиться, и чьи описания интерфейсов являются публично доступными.

сертификат (certificate)

Цифровой или бумажный документ, подтверждающий соответствие между открытым ключом и информацией, идентифицирующей владельца ключа. Он содержит информацию о владельце ключа, сведения об открытом ключе, его назначении и области применения, центре сертификации и т. д. Стандарт X.509 определяет, какая информация должна содержаться в сертификате, и описывает, в каком виде она должна быть в нем записана.

сертификационный центр (Certificate Authority)

Служба, которая выпускает сертификаты. В гриде могут признаваться сертификаты, выпускаемые несколькими сертификационными центрами.

синхронный (synchronous)

О взаимодействии говорится как о синхронном, когда участвующие в нем агенты должны быть доступны для получения и обработки сообщений с момента начала взаимодействия и до тех пор, пока не будут получены все сообщения или пока не возникнет какая-то аварийная ситуация.

состояние (state)

Набор атрибутов, представляющих свойства компоненты в некоторый момент времени.

транзакция (transaction)

Группа последовательных операций, которая представляет из себя логическую единицу работы с данными. Транзакция может быть выполнена целиком либо успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще и тогда она не должна произвести никакого эффекта.

удостоверения (credentials)

Данные, которые передаются для установления требуемой идентичности элемента системы, чья идентификация должна быть аутентифицирована.

хореография (choreography)

Хореография определяет последовательность и условия обмена сообщениями между многочисленными взаимодействующими независимыми агентами.

Хореография веб-служб касается взаимодействия служб с их пользователями. Любой пользователь веб-службы, автоматический или нет, является клиентом этой службы. В качестве пользователей могут выступать другие веб-службы, приложения или люди. Транзакции между веб-службами и их клиентами должны быть четко определены во время их выполнения и могут состоять из множества отдельных взаимодействий, чья композиция составляет полную транзакцию. Эта композиция, ее протоколы передачи сообщений, интерфейсы, установление последовательности и ассоциированная логика и составляет то, что называется хореографией.

хост (host)

Любое устройство, предоставляющее сервисы формата «клиент-сервер» в режиме сервера по каким-либо интерфейсам и уникально определенное на этих интерфейсах. В более частном случае под хостом понимается любой компьютер, сервер,

маршрутизатор, подключенный к локальной или глобальной сети.

элемент хранения (Storage Element, SE)

Любой ресурс хранения данных, зарегистрированный в информационной подсистеме грида. В нем содержатся файлы, зарегистрированные в службе файлового каталога (каталога реплик). Этот ресурс обеспечивает доступ к удаленным сайтам посредством грид-интерфейса. Элемент хранения может управлять большими массивами на дисках, системами хранения сверхбольшой ёмкости и подобными им системами.

DAG (Direct Acyclic Graph)

Направленный граф без циклов - графическое представление зависимостей грид-заданий; каждая работа - это узел DAG. У каждого узла может быть несколько узлов-предков или узлов-потомков, но граф не может иметь петель.

SOAP

Протокол (набор соглашений), определяющих формат и правила обработки SOAP-сообщений. Эти соглашения касаются взаимодействия между SOAP-узлами, генерирующими и принимающими SOAP-сообщения с целью обмена информацией.

Список литературы

1. Ian Foster and Carl Kesselman (eds), "The Grid, Blueprint for a New computing Infrastructure", Morgan Kaufmann Publishers, Inc., 1998.
I. Foster and C. Kesselman (eds), "The Grid 2: Blueprint for a New Computing Infrastructure", Morgan Kaufmann Publishers, 2004.
2. Open Grid Forum (OGF): <http://www.ogf.org>
3. Хорошей «входной точкой» для ознакомления с общей тематикой грида и различными грид-проектами являются грид-порталы:
<http://www.gridclub.ru>
<http://gridcafe.web.cern.ch>
<http://EnterTheGrid.com>
<http://www-1.ibm.com/grid>
4. В.В. Воеводин, Вл.В.Воеводин, «Параллельные вычисления», СПб.:БХВ-Петербург, 2004.
5. Э. Ньюкомер, «Веб-сервисы. XML, WSDL, SOAP и UDDI», Питер, 2003.
6. Foster, "What is Grid? A three point check",
<http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsGrid.pdf>
7. В. Воеводин и М. Филамофитский, "Суперкомпьютер на выходные", Открытые системы, № 05, 2003, <http://www.osp.ru/os/2003/05/043.htm>
8. Проект Seti@Home: <http://setiathome.ssl.berkeley.edu>; <http://setiathome.ru>
9. Проекты MyGrid/OurGrid: <http://dsc.ufcg.edu.br/ourgrid/mygrid/manual>
10. F. Berman, G. Fox, T. Hey (eds.), "Grid computing. Making the Global Infrastructure a Reality", Wiley, 2003.
11. Грид-проекты: <http://www.opensciencegrid.org>
<http://alien.cern.ch>
<http://www.nordugrid.org>
<http://grid.infn.it>
<http://www.nesc.ac.uk>
12. Проект LCG : <http://lcg.web.cern.ch/LCG>
13. Европейский проект EGEE: <http://www.eu-egee.org>
14. Проект GEANT: <http://www.geant.net>
15. Большой адронный коллайдер (Large Hadron Collider, LHC): <http://lhc-new-homepage.web.cern.ch/lhc-new-homepage>
16. Проект РДИГ: <http://egee-rdig.ru>
17. Проект DataGrid: <http://eu-datagrid.web.cern.ch/eu-datagrid>
18. Проект GILDA: <https://gilda.ct.infn.it>
19. Проект Globus: <http://www.globus.org>
20. Проект Condor: <http://www.cs.wisc.edu/condor>
21. Проект VDT: <http://vdt.cs.wisc.edu>
22. Промежуточное программное обеспечение gLite: <http://glite.web.cern.ch/glite>
23. Окончательные версии документов OGF по грид-технологиям (Grid Final

- Documents, GFDs): <http://www.ogf.org/gf/docs/?final>
24. Foster I., Kesselman C., J. Nick, Tuecke S., «The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration.»
<http://www.globus.org/research/papers/ogsa.pdf>
 25. Czajkowski K., Ferguson D., Foster I., Frey J., Graham S., Maguire T., Snelling D., Tuecke S. «From Open Grid Services Infrastructure to WSResource Framework: Refactoring & Evolution.» http://www.globus.org/wsrp/specs/ogsi_to_wsrp_1.0.pdf
 26. L. Ferreira и др. “Introduction to Grid Computing with Globus”, IBM Redbooks,
<http://www.redbooks.ibm.com/abstracts/sg246895.html>
 27. Портал семантического грида (Semantic Grid Community Portal):
<http://www.semanticgrid.org/>
 28. L. Srinivasan and J. Treadwell, "An overview of service-oriented architecture, Web services and grid computing",
http://devresource.hp.com/drc/technical_papers/grid_soa/index.jsp#ws
 29. О. Таковицкий «Технология Grid Computing», <http://www.bytemag.ru/?ID=601856>
 30. Л. Черняк "Web-сервисы, grid-сервисы и другие", Открытые системы N12/2004
<http://www.osp.ru/text/302/184882/>
 31. Проект DEISA: <http://www.deisa.org>
 32. J. Coomer and C. Chabal, "Introduction to the Cluster Grid",
Part I: <http://www.sun.com/blueprints/0802/816-7444-10.pdf>,
Part II: <http://www.sun.com/blueprints/0902/816-7765-10.pdf>
 33. Ian Foster, “The Holy Grail: Industry-Wide System Mgmt Standards at Last?”
<http://news.taborcommunications.com/nview.jsp?appid=360&j=71355&print=1> (перевод на русский доступен по адресу www.gridclub.ru/library/publication.2006-03-28.7071858474/publ_file/)
 34. Foster и др. "Open Grid Services Architecture (OGSA) v1.0",
<http://www.gridforum.org/documents/GFD.30.pdf>
 35. H. Kishimoto and J. Treadwell, "Defining the grid: a roadmap for OGSA standards v1.0",
<http://www.ogf.org/documents/GFD53.pdf>
 36. Organization for the Advancement of Structured Information Standards (OASIS):
<http://www.oasis-open.org/>
 37. Distributed Management Task Force (DMTF): <http://www.dmtf.org>
 38. H. Zimmermann, “OSI Reference Model -The ISO Model of Architecture for Open Systems Interconnection”, IEEE Transactions on communications, vol. 28, no. 4, April 1980, pp. 425 – 432
 39. А.К. Кирьянов и Ю.Ф. Рябов, “Введение в технологию Грид”, Учебное пособие, ПИЯФ РАН, 2006
 40. Стандарт WSDM: www.oasis-open.org/committees/wsdm
 41. Foster, T. Maguire, D. Snelling, “OGSA WSRF Basic Profile1.0”,
<http://www.ggf.org/documents/GFD.72.pdf>
 42. Д. Талиа, “OGSA: где Grid встречается с Web”, Открытые системы , 2003 , №1, 47-50; <http://www.osp.ru/os/2003/01/182408>
 43. Б. Сандарам, “Что такое WSRF”,

- <http://www.gridclub.ru/library/publication.2005-10-17.1342842216> (англоязычный вариант: <http://www.ibm.com/developerworks/edu/gr-dw-gr-wsrfl-i.html>); Джудит Майерсон, "Работа с Web-сервисами в корпоративных SOA: Часть 8. Извещение Web-сервисов о наличии в бизнес-системе EAI разнородных SOA", <http://www.ibm.com/developerworks/ru/library/ws-soa-enter8/index.html>
44. GSI: <http://forge.gridforum.org/projects/gsi-wg>
 45. CORBA: <http://www.omg.org>
 46. DCOM: <http://msdn.microsoft.com/library/en-us/dnanchor/html/dcom.asp>
 47. Internet Communications Engine (ICE): <http://www.zeroc.com>
 48. Менеджер ресурсов хранения данных (SRM): <http://sdm.lbl.gov/srm-wg>
 49. Andrew File System (AFS): <http://www.openafs.org>
 50. Проект Open Science Grid (OSG): <http://www.opensciencegrid.org/>
 51. Проект AliEn: <http://alien.cern.ch>
 52. Проект Nordugrid: <http://www.nordugrid.org>
 53. В. Tierney и др. "A Grid Monitoring Architecture" , <http://www.ogf.org/documents/GFD7.pdf>
 54. Relational Grid Monitoring Architecture (R-GMA): <http://www.r-gma.org>
 55. Portable Batch System (PBS): <http://www.openpbs.org>
 56. Load Sharing Facility (LSF): <http://www.platform.com/Products;>
<http://wwwpdp.web.cern.ch/wwwpdp/bis/services/lsf>
 57. American National Standards Institute (ANSI): <http://ansi.org>,
International Organization for Standardization (ISO): <http://www.iso.org>
 58. Стандарт X.509: <http://www.itu.int/rec/T-REC-X.509-200508-I>
 59. International Telecommunication Union (ITU): <http://www.itu.int>
 60. Ричард Э. Смит, «Аутентификация: от паролей до открытых ключей», Вильямс, 2002 г., 432 стр.
 61. Н. Смарт, «Криптография», Техносфера, 2006 г., 528 стр.
 62. "Глоссарий терминов, принятых в OSG"
[http://www.gridclub.ru/library/publication.2006-02-28.3233208252/view;](http://www.gridclub.ru/library/publication.2006-02-28.3233208252/view)
"Глоссарий Web-служб" <http://www.gridclub.ru/library/publication.2005-10-17.1036125416/view>
 63. Википедия: <http://ru.wikipedia.org>

Андрей Павлович Демичев,
Вячеслав Анатольевич Ильин,
Александр Павлович Крюков

Введение в грид-технологии

Препринт НИИЯФ МГУ - 2007 - 11/832

Работа поступила в ОНТИ НИИЯФ МГУ 29.10.2007