

УДК 004.032.24

**А.А. АЛЬ-ХУЛАЙДИ**

## **ПРОГРАММНЫЕ ПАКЕТЫ ОБЕСПЕЧЕНИЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ**

*Развитие технологий Grid подошло к фазе реализации проектов, направленных на развертывание крупных действующих инфраструктур. Использование их на практике поставило не только новые задачи, но и вызвало необходимость в оценке подходов, которые были разработаны и успешно применялись в пробных экспериментах. Одной из главной составляющей таких инфраструктур является программное обеспечение, которое рассмотрено в данной статье.*

**Ключевые слова:** Grid, ресурсная архитектура, кластер, безопасность, диспетчеризация.

**Введение.** В последние годы за рубежом и в нашей стране активно ведутся работы, связанные с объединением глобальных вычислительных ресурсов в единую распределенную систему вычислений, хранения и передачи информации. Такая система (Grid) обеспечивает доступ пользователей к миллионам географически распределенных компьютерных ресурсов.

В настоящее время уже создана техническая база (телекоммуникационные сети большой пропускной способности) и сделаны шаги по созданию программного обеспечения для работы в среде Grid. Созданы такие программные пакеты, как Globus Toolkit, ARC- Grid, DataGrid, UNICORE Grid, которые предлагают базовые средства для создания Grid-инфраструктуры: средства обеспечения безопасности в распределенной среде, средства надежной передачи больших объемов данных, средства запуска и получения результатов выполнения задач на удаленных вычислительных ресурсах.

**Задачи исследования.** Перед тем, как мы перейдем к рассмотрению и сравнению данных программных проектов, дадим определение понятия «grid-система». Grid (вычислительная решётка) - это распределенная информационно-вычислительная среда, построенная на основе Всемирной паутины (World WideWeb), которая обеспечивает масштабируемые, безопасные, быстродействующие механизмы для обнаружения и доступа к удаленным вычислительным и информационным ресурсам [1].

В отличие от кластерных систем, которые являются частным случаем grid, мировое сообщество исследователей и разработчиков grid-систем рассматривают их как единое целое и не выделяют параметры классификации и не проводят самой классификации grid. Но если провести аналогии с кластерными системами и проанализировать вопрос, то можно определить, что данные системы можно разделить на следующие три типа [2]:

1. GRID на основе использования добровольно предоставляемого свободного ресурса персональных компьютеров («добровольная» GRID);
2. Научная GRID – хорошо распараллеливаемые приложения программируются специальным образом (например, с использованием Globus Toolkit);
3. GRID на основе выделения вычислительных ресурсов по требованию (Enterprise GRID или коммерческая GRID); обычные коммерческие приложения работают на виртуальном компьютере, который, в свою очередь, состоит из нескольких физических компьютеров, объединённых с помощью GRID-технологий (кластер).

Принципы функционирования Grid-системы представлены на рис.1.



Рис.1. Принцип функционирования Grid-систем

Таким образом, к **прикладным задачам**, при решении которых можно использовать GRID, относятся:

- сложное моделирование;
- совместная визуализация очень больших наборов научных данных;
- распределенная обработка для анализа данных;
- связь научного инструментария с удаленными компьютерами и архивами данных.

Наиболее эффективно применение распределённых вычислительных систем для решения следующих **задач**:

- распределенные высокопроизводительные вычисления, решение очень больших задач, требующих максимальных процессорных ресурсов, памяти и т.д.;
- «высокопоточные» вычисления, позволяющие организовать эффективное использование ресурсов для небольших задач, утилизируя временно простаивающие компьютерные ресурсы;
- проведение крупных разовых расчетов;
- вычисления с привлечением больших объемов распределенных данных, например, в метеорологии, астрономии, физике высоких энергий;

- коллективные вычисления: одновременная работа нескольких взаимодействующих пользователей.

На рис.2 показан пример вычислительной Grid-сети.

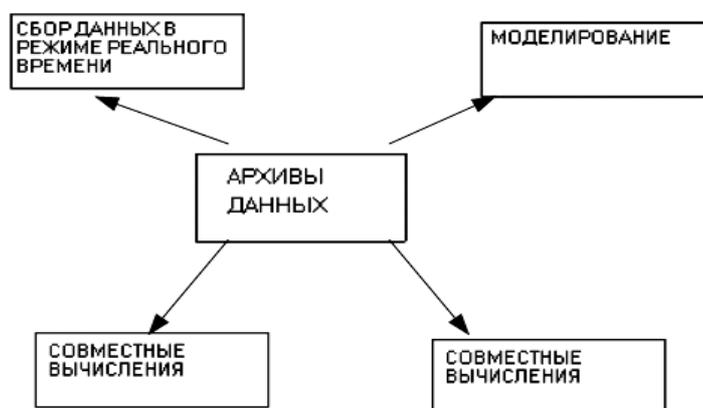


Рис.2. Пример вычислительной Grid-сети

**Обзор существующих пакетов Grid-систем.** Разрабатываемые в настоящее время программные системы делятся на два основных типа: свободно распространяемые и системы коммерческого типа. Проведем сравнительный анализ программных пакетов Grid-систем этих типов.

С развитием технологии Grid-систем стали появляться не отдельные ПО, а программные платформы, то есть согласованные наборы средств. Данные платформы со значительно большей полнотой по сравнению с пакетами ПО поддерживают развертывание и обслуживание Grid, а также организуют его функционирование как единой операционной среды. Характерные отличия платформ проявляются в том, что в них в рамках единой архитектуры начинают решаться задачи обеспечения надежности, детерминированности, качества обслуживания, управляемости Grid с помощью механизмов планирования, мониторинга заданий и устройств, учета и протоколирования.

К складывающемуся классу платформ можно отнести два европейских комплекса средств: **DataGrid** [3] и **Unicore** [4]. Первый из этих комплексов, разработанный в течение 2000 - 2003 годов, представляет для нас особый интерес, поскольку он становится стартовой программной базой проекта EGEE, в котором принимает участие и ИПМ РАН. Рассмотрение ПО DataGrid позволяет сделать заключение, что оно способно обеспечить функционирование Grid большого масштаба, но с определенными ограничениями по способу организации ресурсов и режиму их использования.

**Функциональные возможности.** Ядро ПО DataGrid составляет система WMS (Workload Management System), в которую входят следующие основные компоненты:

- User Interface. Пользовательский интерфейс на основе языка JDL;
- Resource Broker. Выполняет подбор ресурсов для исполнения запросов.
- CondorG. Выполняет удаленные операции по управлению заданиями на ресурсах.
- Logging and Bookkeeping. Службы протоколирования событий, генерируемых компонентами WMS и учета использования ресурсов.
- MyProxy. Служба, поддерживающая обновление доверительных сертификатов, используется для заданий с длительным выполнением.
- DataGrid Accounting System. Поддерживает экономические отношения между поставщиками и потребителями ресурсов.

Как видно из этого списка, DataGrid решает важнейшие для распределенной среды задачи, обеспечивая виртуализацию ресурсов, протоколирование, учет, поддержку отношений пре-

доставления/потребления ресурсов в рамках общей архитектуры взаимодействующих между собой служб.

Обслуживаемый класс приложений. Основной класс – однопроцессорные вычислительные задания с большим временем счета. Поскольку в DataGrid помимо WMS интегрированы служба GridFTP и служба репликации файлов Replica Location Service, платформа способна обеспечивать работу с файлами большого объема.

**Характеристика ресурсов.** DataGrid рассчитана на кластеризованные ресурсы с любыми системами управления, с которыми имеет интерфейс Globus Toolkit. Способ выбора ресурсов для заданий основан на информации о количестве заданий в очередях кластерных систем управления и способен дать хорошие результаты при условии, что ресурсы функционируют в выделенном режиме, то есть полностью отчуждаются от владельцев. DataGrid использует информационную службу Globus Toolkit для распределенного мониторинга ресурсов.

**Надежность обработки.** Гарантируется завершение обработки запущенных заданий. Мониторинг задания и перезапуск в аварийных ситуациях осуществляется из точки запуска с помощью системы CondorG. Имеется аппарат прикладных контрольных точек, однако, автоматический рестарт не поддерживается.

**Качество обслуживания.** Способ планирования в DataGrid не позволяет получить точную оценку времени обработки заданий и тем самым качество обслуживания не регулируется.

**Масштабируемость.** Распределение заданий производится по централизованной схеме, оценки производительности серверов пока отсутствуют.

Платформа **UNICORE** имеет несколько особенностей, обусловленных тем, что с самого начала она предназначалась для обеспечения доступа к суперкомпьютерным центрам стран Центральной Европы. Unicore – одна из немногих оригинальных платформ, которая не использовала Globus Toolkit, и лишь в последнее время происходит трансформация ее протоколов в сторону общепризнанных для Grid.

Унифицированный интерфейс с вычислительными ресурсами - **Unicore** (UNiform Interface to COmputing REsources) предоставляет ученым и инженерам ресурсы суперкомпьютерных центров, объединенных в Grid, и делает их доступными через Интернет. Среда Grid UNICORE поддерживает высокий уровень безопасности, а аутентификация осуществляется совместимым между ресурсами и прозрачным для пользователей способом. Различия между платформами полностью скрыты, так что Unicore можно рассматривать как портал, открывающий бесшовный дистанционный доступ к суперкомпьютерам, компиляции и выполнению приложений и пересылке данных ввода-вывода.

Выигрыш, который получает пользователь UNICORE, - однородный доступ к разного рода системам, а значит, и больше возможностей по получению ресурсов. Вычислительные центры, применяющие UNICORE, смогут уменьшить количество документации для пользователей, а также более эффективно использовать свои вычислительные ресурсы.

Главные компоненты UNICORE:

- агент подготовки заданий (JPA);
- контроллер монитора заданий (JMC);
- сервер https UNICORE, также называемый шлюзом (Gateway);
- супервизор сетевых заданий (NJS);
- графический интерфейс пользователя, основанный на Java-апплетах.

Клиент UNICORE дает возможность потребителю создавать, представлять и управлять заданиями с любой рабочей станции или персонального компьютера, подключенных к Internet. Клиент соединяется с UNICORE через шлюз, который подтверждает подлинность клиента и пользователя перед контактированием с серверами UNICORE, которые, в свою очередь, управляют представленными UNICORE-заданиями. Задания, предназначенные для локальных компьютеров, выполняются на их пакетных подсистемах, те же, которые будут выполнены на удаленных узлах, передаются на просмотр шлюзу. Все необходимые передачи данных и синхронизации выполняют

ся серверами. Серверы также сохраняют информацию о статусе и результатах работы, передавая их клиентам по запросу пользователя. Компоненты третьей стороны, такие как Globus, могут быть интегрированы в структуру UNICORE для расширения его функциональности.

Свободно распространяемая платформа **ARC** (Advanced Resource Connector) разработана для проекта стран северной Европы (NordGrid) и ориентирована на поддержку приложений физики высоких энергий (эксперименты LHC).

Основные архитектурные решения ARC следуют общепринятым подходам построения Grid. Используется организация ресурсов, аналогичная применяемой в EU DataGrid. Узлами инфраструктуры служат кластеры, управляемые системами пакетной обработки, или отдельные компьютеры. Узлы комплектуются элементами памяти (Storage Elements).

ARC обеспечивает следующие функции:

- информационная;
- включение ресурсов в Grid и их мониторинг;
- запуск и управление заданиями;
- брокеринг (распределение заданий по ресурсам);
- управление данными и ресурсами.

Все функции реализованы в виде служб, которые опираются на известные программные средства с открытым кодом OpenLDAP, OpenSSL и SASL. Реализация выполнена с помощью библиотек Globus Toolkit 2 (GT2), безопасность достигается путем использования протоколов и инфраструктурных решений GSI.

Отличительной особенностью NordGrid ARC является то, что хотя эта платформа и опирается на протоколы GT2 и реализована посредством API GT2, в ней предложен собственный набор служб, заменяющий службы GT2. ARC не использует GRAM, утилиты управления заданиями, Gatekeeper и скрипты Job-manager, сервер Wuftp, схемы и информационные поставщики MDS. Для всего этого предложены собственные варианты:

- Grid Manager;
- gridftpd (ARC/NordGrid GridFTP server);
- user Interface;
- Broker ("персональный" брокер, встроенный в пользовательский интерфейс);
- система мониторинга.

Кроме того, определена новая информационная схема и для нее разработаны поставщики данных, расширен язык описания ресурсов (xRSL). Таким образом, платформа ARC, хотя и построена на библиотеках GT2, это совсем не GT2, она имеет свой набор служб.

На ресурсном узле работают следующие программы ARC: Grid Manager, Gridftpd и информационные агенты. Служба Grid Manager локально запускает задание и контролирует процесс выполнения. Служба Gridftpd осуществляет дистанционный прием заданий, создавая для каждого отдельный каталог на все время обработки. На узлах работают также поставщики информации о состоянии ресурсов, собирающие данные и передающие их индексным службам, которые представляют собой упрощенный вариант базы данных GT2 GIIS и могут быть связаны в произвольную сетку: по связям происходит динамическое размножение информации. К индексным службам обращается пользовательский интерфейс (например, для выбора исполнительного узла) и Grid Manager.

Дистанционный запуск заданий производится посредством пользовательского интерфейса (UI), который представляет собой набор команд для запуска, мониторинга и управления заданиями, а также перемещения файлов и опроса информации о состоянии ресурсов. В состав UI входит Broker, функция которого – подбор наиболее подходящего ресурса для задания. Другой специальный клиент - Grid Monitor через любой Web-браузер периодически «опрашивает» распределенную информационную систему, представляя результаты в виде взаимосвязанных страниц.

При разработке ARC преследовалась цель создания ПО для производства, используя в качестве основного принципа максимально полную децентрализацию, поэтому на каждом рабочем месте пользователя Grid устанавливается персональный брокер, независимо подбирающий ресурсы для запускаемых заданий. Этот подход отличен от централизованной схемы EU DataGrid с единственным брокером на все рабочие места.

Инструментарий с открытым кодом **Globus Toolkit** был разработан Калифорнийским университетом и Аргоннской национальной лабораторией при поддержке ведущих компаний США и Японии. Отдельно следует отметить вклад корпорации IBM, которая участвует в проекте Globus и выпускает свой инструментальный GRID Toolbox, построенный на основе Globus Toolkit.

Globus Toolkit – это инфраструктура стандартов и инструментов. Основным стандартом в рамках проекта Globus является OGSA (Open Grid Services Architecture) - открытая архитектура вычислений учрежденческого уровня. Архитектура OGSA, созданная на концепциях и технологиях, разработанных специалистами в области Grid и Web-служб, определяет единую семантику представления служб, стандартные механизмы для создания, именованного обнаружения экземпляров Grid-служб, обеспечивает прозрачность местонахождения и связывания различных протоколов и поддерживает интеграцию с базовыми механизмами нижележащих платформ. Разработка технической спецификации OGSA ведется в рамках форума Grid Global Forum. Пакет Globus Toolkit доступен для загрузки из Интернет в виде исходных текстов или дистрибутивов. IBM предоставляет GRID Toolbox в виде дистрибутивов, которые адаптированы под платформы pSeries, zSeries и xSeries. Глобус (The Globus Toolkit™) основан на четырех основных протоколах:

- Уровень связи:  
защищенность: Grid Security Infrastructure (GSI).
- Уровень ресурсов:  
управление ресурсами: Grid Resource Allocation Management (GRAM).
- Информационный сервис: Grid Resource Information Protocol (GRIP).
- Пересылка данных: Grid File Transfer Protocol (GridFTP).

Заслуживает внимания появление, наряду со свободно распространяемыми платформами, целого ряда коммерческих продуктов для глобально распределенных вычислений: **DCGrid, LiveCluster, GridMP, Frontier**, выпущенных соответственно компаниями Entropia, DataSynapse, United Devices и Paragon Computation. Эти продукты квалифицируются производителями как платформы для Grid, и с этим, в общем, трудно не согласиться, поскольку они направлены на решение задач Grid (обеспечение скоординированного доступа к глобально распределенным ресурсам в рамках виртуальной организации) и поддерживают полный цикл обработки заданий в распределенной инфраструктуре.

Применяемый в коммерческих платформах подход имеет ряд особенностей. Прежде всего это касается организации ресурсной составляющей Grid. В платформах, основанных на Globus Toolkit, узлы Grid являются многомашинными комплексами, которые находятся в автономном административном домене, связаны локальной сетью и управляются системой пакетной обработки, играющей роль менеджера ресурсов (MP). В коммерческих же платформах узлы могут быть элементарными, то есть каждый узел – это компьютер, на который для подключения в Grid устанавливается компактный агент, выполняющий функции запуска заданий, мониторинга, защиты и контактов с управляющим центром (Grid-сервером).

Различие в типах ресурсов сказывается на архитектуре поддерживаемых Grid. Коммерческие платформы можно назвать вертикально интегрированными из-за того, что в них узлы - глобально распределенные компьютеры – недоступны для прямого запуска заданий, а интегрируются через управляющий центр, который, с одной стороны, представляет собой точку доступа ко всем ресурсам, а, с другой, выполняет функции MP, управляя ресурсами и виртуализируя их. Можно провести аналогию вертикально интегрированного Grid с кластерной архитектурой, но со следующим важным отличием: обрабатывающие узлы могут быть глобально распределены и принадлежать разным владельцам.

В коммерческих платформах предложен ряд новых программных решений, направленных на поддержку такого режима использования ресурсов, когда они не отчуждаются от владельцев. В этих условиях задачи Grid усложняются, так как ресурсная составляющая не является ни надежной (машины могут выключаться и перезагружаться), ни безопасной (посторонние лица могут иметь дистанционный или прямой доступ к машине), ни предсказуемой (в любой момент машина может быть полностью занята владельцем). В связи с этими проблемами разработаны специальные методы. Безопасность, например, рассматривается в трех аспектах: защита владельца ресурса, защита конфигурации машины, защита приложения Grid, и обеспечивается механизмами "изоляции" приложений в виртуальной исполнительской среде.

Два вида платформ с вертикальной интеграцией (коммерческие платформы) и с горизонтальной интеграцией на основе стандартов дистанционного взаимодействия в архитектуре Grid-служб OGSA, можно, наверное, рассматривать как несводимые альтернативы. Однако более продуктивным представляется взаимное сближение на основе как межплатформенной интероперабельности, так и интеграции конкретных механизмов и функций. Фактически сближение уже началось, это проявляется, например, в том, что перечисленные выше компании-разработчики в той или иной степени принимали участие в подготовке стандарта OGSA и имеют планы перевести свои частные протоколы взаимодействия в архитектуру Grid-служб.

#### Характеристики программных пакетов

Программные пакеты	Характеристика
DataGrid	Предоставляет конечному пользователю прозрачный доступ к пространственно распределенным вычислительным ресурсам и хранилищам данных, организованным в виде независимых сайтов. Структура сайтов унифицирована. DataGrid рассчитана на кластеризованные ресурсы с любыми системами управления, с которыми имеет интерфейс Globus Toolkit. Способ выбора ресурсов для заданий основан на информации о количестве заданий в очередях кластерных систем управления.
Globus Toolkit	Неготовое решение, это инфраструктура стандартов и инструментов. Обеспечивает прозрачность местонахождения и связывания различных протоколов и поддерживает интеграцию с базовыми механизмами нижележащих платформ
ARC- Grid	Узлами инфраструктуры служат кластеры, управляемые системами пакетной обработки, или отдельные компьютеры. Узлы комплектуются элементами памяти. Использует в качестве основного принципа максимально полную децентрализацию, поэтому на каждом рабочем месте пользователя Grid устанавливается персональный брокер, независимо подбирающий ресурсы для запускаемых заданий. Этот подход отличен от централизованной схемы DataGrid с единственным брокером на все рабочие места.
UNICORE Grid	Единый и легкий в использовании графический интерфейс пользователя, открытая архитектура, основанная на понятии абстрактного задания, целостная архитектура защиты, минимальная зависимость от локальных административных процедур, эксплуатация существующих и находящихся на стадии становления технологий, основанных на стандарте Java и Web-технологиях.

**Заключение.** Большая часть программных систем для Grid, хотя и разрабатывалась различными коллективами, основывалась на стандартах Globus Toolkit (эволюционировавших в архитектуру Grid-служб OGSA), и вследствие этого они оказались хорошо совместимыми друг с другом. Реализованный в Globus Toolkit механизм поддержки контекста безопасности, основанный на доверительной передаче прав доступа, рассчитан на произвольное количество шагов обработки задания. Оправданность и предусмотрительность такого решения была доказана при реализации всех систем серверного обслуживания, основанных на Globus Toolkit.

#### **Библиографический список**

1. Foster I., Kesselman C., Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications. <http://www.globus.org/research/papers/anatomy.pdf>.
2. 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>
3. Graham S., Simeonov S., Boubez T., Daniels G., Davis D., Nakamura Y., Neyama R.. Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI, 2001.
4. Graupner S., Pruyne J., Singhal S. Making the Utility Data Center a Power Station for the Enterprise Grid, HP Laboratories Palo Alto. <http://www.hpl.hp.com/techreports/2003/HPL-2003-53.pdf>

Материал поступил в редакцию 02.11.09.

#### **AL-KHULAI DI ABDULMAJED AHMED**

#### **SOFTWARE PACKAGES FOR PARALLEL COMPUTING**

Grid development has approached the phase of realization of projects aimed at deployment of the large-scale operating infrastructures. Reality of practical implementation has not only brought in a whole new set of tasks, but also demands reevaluation of the methods that were previously developed and successfully tested in pilot experiments. One of the main component of such infrastructure is the software discussed in this article.

**Аль-хулайди Абдулмаджид Ахмед** (р.1983), аспирант кафедры «Программное обеспечение вычислительной техники и автоматизированных систем» ДГТУ. Окончил ДГТУ (2008).

Научные интересы: вычислительные сети, распределенные вычислительные системы, анализ и синтез телекоммуникационных сетей.

Автор 4 научных статей.

[alkhulaidi\\_2006@hotmail.com](mailto:alkhulaidi_2006@hotmail.com)