

11. Технологии Грид

Термин «грид» был введен в обращение Яном Фостером в начале 1998 года публикацией книги «Грид. Новая инфраструктура вычислений» [27]:

Грид – это система, которая координирует распределенные ресурсы посредством стандартных, открытых, универсальных протоколов и интерфейсов для обеспечения нетривиального качества обслуживания (QoS – Quality of Service).

Хотя в последнее десятилетие базовая идея грид не претерпела существенных изменений, всеобъемлющего определения грид не существует до сих пор [62].

11.1 Архитектура Грид

Основной идеей, заложенной в концепции грид-вычислений, является централизованное удаленное предоставление ресурсов, необходимых для решения различного рода вычислительных задач. В каком-то смысле, концепция грид-вычислений идея рифмуется с концепцией электросети (англ. Power Grid): нам не важно, откуда к нам в розетку приходит электричество. Независимо от этого мы можем подключить к электросети утюг, компьютер или стиральную машину. Также и в идеологии грид: мы можем запустить любую задачу с любого компьютера или мобильного устройств на вычисление, ресурсы же для этого вычисления должны быть автоматически предоставлены на удаленных высокопроизводительных серверах, независимо от типа нашей задачи.

С более практической точки зрения, основная задача, лежащая в основе концепции грид, это согласованное распределение ресурсов и решение задач в условиях динамических, многопрофильных *виртуальных организаций*. Распределение ресурсов, в котором заинтересованы разработчики грид, это не обмен файлами, а прямой доступ к компьютерам, программному обеспечению, данным и другим ресурсам, которые требуются для совместного решения задач и стратегий управления ресурсами, возникающих в промышленности, науке и технике. *Виртуальной организацией (ВО)* называют ряд отдельных людей или учреждений, объединенных едиными правилами коллективного доступа к распределенным вычислительным ресурсам [31]. Для организации работы в рамках таких ВО возникает необходимость в следующем:

1. в гибких механизмах разделения ресурсов, начиная от клиент-серверных заканчивая одноранговыми;
2. в развитой системе контроля используемых ресурсов, включая контроль над мелкомодульными и другими методами доступа и использование локальных и глобальных подходов;
3. в распределенном доступе к различным ресурсам, начиная от программ, файлов и данных заканчивая компьютерами, сенсорами и сетями;
4. в различных моделях использования ресурсов (от однопользовательских до многопользовательских, от высокопроизводительных до мало затратных) и, следовательно, включающих регулирование качества предоставляемого обслуживания, планирование, перераспределение и ведение учета ресурсов.

Анализ альтернативных технологий построения распределенных вычислительных систем, проведенный в [31], показал, что их применение не позволяет в полной мере достичь исполнения всех требований, указанных выше. В соответствии с этим была предложена альтернативная архитектура грид. Исследования и разработки в сообществе грид привели к разработке протоколов, сервисов и инструментария, направленного именно на те проблемы, которые возникают при попытке создания масштабируемых ВО. Эти технологии включают в себя:

1. решения по безопасности, поддерживающие управление сертификацией и политиками безопасности, когда вычисления производятся несколькими организациями;
2. протоколы управления ресурсами и сервисами, поддерживающие безопасный удаленный доступ к вычислительным ресурсам и ресурсам данных, а также перераспределение различных ресурсов;
3. протоколы запроса информации и сервисы, обеспечивающие настройку и мониторинг состояния ресурсов, организаций и сервисов;
4. сервисы обработки данных, обеспечивающие поиск и передачу наборов данных между системами хранения данных и приложениями.

Выделяют следующие уровни архитектуры грид:

1. *Базовый уровень (Fabric)* – содержит различные ресурсы, такие как компьютеры, устройства хранения, сети, сенсоры и др.
2. *Связывающий уровень (Connectivity)* – определяет коммуникационные протоколы и протоколы аутентификации.
3. *Ресурсный уровень (Resource)* – реализует протоколы взаимодействия с ресурсами РВС и их управления.

4. *Коллективный уровень (Collective)* – управление каталогами ресурсов, диагностика, мониторинг;
5. *Прикладной уровень (Applications)* – инструментарий для работы с грид и пользовательские приложения.

На *базовом уровне* определяются службы, обеспечивающие непосредственный доступ к ресурсам, использование которых распределено посредством протоколов Грид.

1. 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.
2. Ресурсы памяти представляют собой пространство для хранения данных. Для доступа к ресурсам памяти также используется программное обеспечение промежуточного уровня, реализующее унифицированный интерфейс управления и передачи данных.
3. Информационные ресурсы и каталоги являются особым видом ресурсов памяти. Они служат для хранения и предоставления метаданных и информации о других ресурсах Грид-системы.
4. Сетевой ресурс является связующим звеном между распределенными ресурсами Грид-системы. Основной характеристикой сетевого ресурса является скорость передачи данных.

Связывающий уровень определяет коммуникационные протоколы и протоколы аутентификации, обеспечивая передачу данных между ресурсами базового уровня. Связывающий уровень грид основан на стеке протоколов TCP/IP:

1. Интернет (IP, ICMP);
2. Транспортные протоколы (TCP, UDP);
3. Прикладные протоколы (DNS, OSRF...).

Ресурсный уровень реализует протоколы, обеспечивающие выполнение следующих функций:

- согласование политик безопасности использования ресурса;
- процедура инициации ресурса;
- мониторинг состояния ресурса;
- контроль над ресурсом;
- учет использования ресурса.

Отдельно выделяются 2 типа протоколов ресурсного уровня:

1. *Информационные протоколы* – используются для получения информации о структуре и состоянии ресурса.
2. *Протоколы управления* – используются для согласования доступа к разделяемым ресурсам, определяя требований и допустимых действий по отношению к ресурсу (например, поддержка резервирования, возможность создания процессов, доступ к данным).

Коллективный уровень отвечает за глобальную интеграцию различных наборов ресурсов и может включать в себя службы каталогов; службы совместного выделения, планирования и распределения ресурсов; службы мониторинга и диагностики ресурсов; службы репликации данных.

На *прикладном уровне* располагаются пользовательские приложения, исполняемые в среде ВО. Они могут использовать ресурсы, находящиеся на любых нижних слоях архитектуры Грид.

11.2 Стандарты Грид

Ключевым моментом в разработке грид приложений является *стандартизация*, позволяющая организовать поиск, использование, размещение и мониторинг различных компонентов, составляющих единую виртуальную систему, даже если они предоставляются различными поставщиками услуг или управляются различными организациями [28]. К началу 2001 года в различных проектах были представлены различные методы реализации грид-вычислений. Но все они сходились в одном: для гибкого, прозрачного и надежного предоставления доступа к вычислительным ресурсам была предложена сервисно-ориентированная модель.

В 2001 году в качестве базы для создания стандарта архитектуры грид приложений была выбрана технология веб-сервисов. Данный выбор был обусловлен двумя основными достоинствами данной технологии. Во-первых, язык описания интерфейсов веб-сервисов WSDL (Web Service Definition Language) поддерживает стандартные механизмы для определения интерфейсов отдельно от их реализации, что в совокупности со специальными механизмами связывания (транспортным протоколом и форматом кодирования данных) обеспечивает возможность динамического поиска и компоновки сервисов в гетерогенных средах. Во-вторых, широко распространенная адаптация механизмов веб-сервисов означает, что инфраструктура, построенная на базе веб-сервисов, может использовать различные утилиты и другие существующие сервисы, такие

как различные процессоры WSDL, системы планирования потоков задач и среды для размещения веб-сервисов [30].

Разработанный стандарт архитектуры грид получил название *OGSA* (Open Grid Services Architecture – Открытая архитектура грид-сервисов) [18]. Он основывается на понятии грид-сервиса. *Грид-сервисом* называется сервис, поддерживающий предоставление полной информации о текущем состоянии (потенциально временного) экземпляра сервиса, а также поддерживающий возможность надежного и безопасного исполнения, управления временем жизни, рассылки уведомлений об изменении состояния экземпляра сервиса, управления политикой доступа к ресурсам, управления сертификатами доступа и виртуализации [30]. Грид-сервис поддерживает следующие стандартные интерфейсы.

1. *Поиск*. Грид приложениям необходимы механизмы для поиска доступных сервисов и определения их характеристик.
2. *Динамическое создание сервисов*. Возможность динамического создания и управления сервисами – это один из базовых принципов OGSA, требующий наличия сервисов создания новых сервисов.
3. *Управление временем жизни*. Распределенная система должна обеспечивать возможность уничтожения экземпляра грид-сервиса.
4. *Уведомление*. Для обеспечения работы грид приложения наборы грид сервисов должны иметь возможность асинхронно уведомлять друг друга о изменениях в их состоянии.

Первая реализация модели OGSA, разработанная в 2003 г., называлась *OGSI* (Open Grid Service Infrastructure). В связи с тем, что существовавшие тогда стандарты веб-сервисов (к которым относились WSDL, SOAP, UDDI) не могли обеспечить всех требований, предъявляемых разработчиками к функциональным возможностям грид-сервисов, при создании OGSI потребовалось модифицировать и расширить соответствующие стандарты [21]. Это привело к тому, что совместное использование веб-сервисов и грид-сервисов в одной среде стало невозможным, из-за несовместимости базовых стандартов [4].

Дальнейшие совместные усилия сообщества грид и организаций по разработке стандартов веб-сервисов привело к определению стандартов, соответствующих требованиям грид. В предложенном стандарте *WSRF* (*Web Service Resource Framework*) [13, 19, 22, 66] специфицированы универсальные механизмы для определения, просмотра и управления состоянием удаленного ресурса, что является критически-важным с точки зрения грид [25]. На сегодняшний день реализация модели OGSA посредством стандарта WSRF (и сопут-

ствующих стандартов, таких как WS-Notification и WS-Addressing) является наиболее распространенной в среде грид.

В настоящее время, существуют две системы, обеспечивающие инфраструктуру разработки грид-систем в соответствии со стандартами OGSA, реализованными посредством WSRF: Globus Toolkit [20] и UNICORE [48].

11.3 Система Globus

Globus – это проект по разработке и предоставлению инфраструктуры для грид-вычислений [20]. Становление данного проекта приходится на 1997 год, а его разработка продолжается и сегодня. Когда как первоначально Globus был развитием проекта I-WAY, в процессе развития, основной акцент был перенесен с поддержки высокопроизводительных вычислений в сторону сервисов поддержки виртуальных организаций.

Цель его создания – предоставление возможности приложениям работать с распределенными разнородными вычислительными ресурсами как с единой виртуальной машиной. Основная направленность данного проекта – вычислительные грид-системы. Под вычислительной грид-системой подразумевается инфраструктура аппаратных и программных ресурсов, реализующая надежный и полномасштабный доступ к высокопроизводительным вычислительным системам, независимо от географического расположения пользователей или ресурсов.

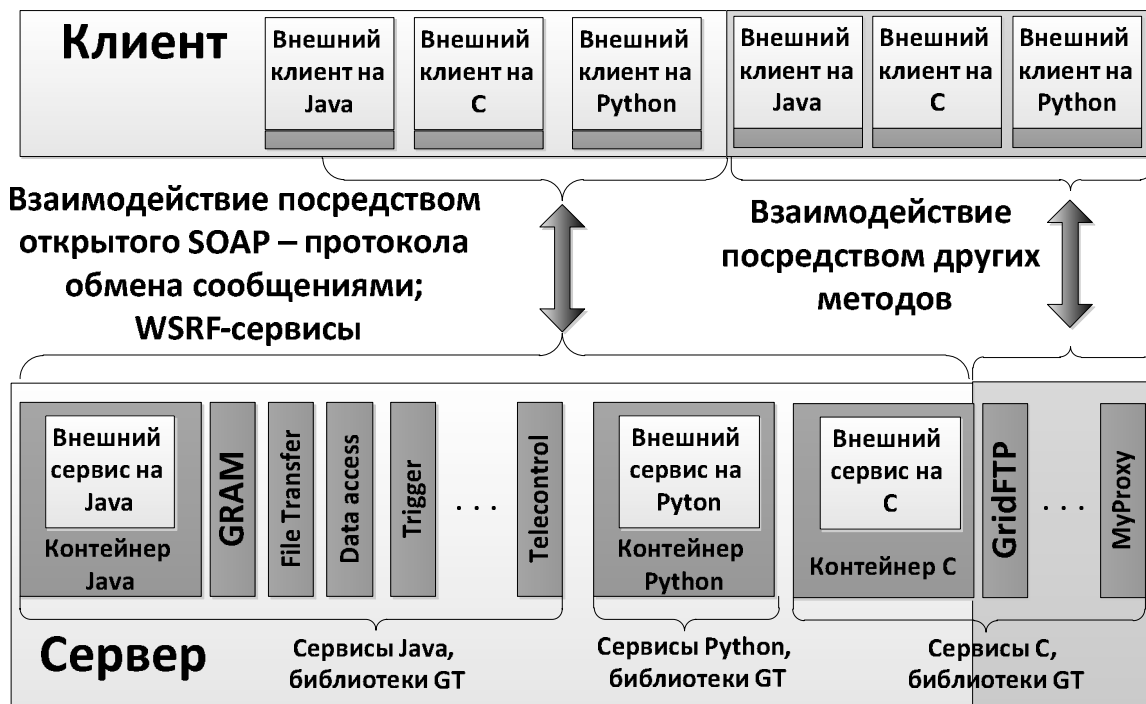


Рис. 58. Общая схема взаимодействия компонентов Globus Toolkit 4.0

Базовым элементом системы Globus выступает Globus Toolkit (инструментарий Globus), описывающий базовые сервисы и возможности, необходимые для создания вычислительных грид-систем [20]. Система Globus предоставляет

высокоуровневым приложениям доступ к сервисам, каждый из которых приложение или разработчик может использовать для достижения собственных целей. Такой метод работы может быть реализован только при высокой степени изолированности отдельных сервисов и четко определенном программном интерфейсе каждого предоставляемого сервиса.

Рассмотрим базовые сервисы, предоставляемые системой Globus на сегодняшний день (см. рис. 58).

1. Протокол GRAM (“Globus Toolkit Resource Allocation Manager” – Менеджер Распределения Ресурсов Globus Toolkit) используется для распределения вычислительных ресурсов и для контроля вычислений, с использованием данных ресурсов.
2. Расширенная версия протокола передачи файлов GridFTP используется для организации доступа к данным, включая вопросы безопасности и параллелизма высокоскоростной передачи данных.
3. Контейнеры для пользовательских сервисов, поддерживающие аутентификацию, управление состоянием, поиск и т.п. обеспечивающие поддержку стандартов WSRF, WS-Security, WS-Notification.
4. Сервисы аутентификации и безопасности соединений GSI (“Grid Security Infrastructure” – Инфраструктура Безопасности Грид).
5. Распределенный доступ к информации о структуре и состоянии системы распределенных вычислений.
6. Удаленный доступ к данным посредством последовательных и параллельных интерфейсов.
7. Создание, кэширование и поиск исполняемых ресурсов.
8. Библиотеки, для обеспечения взаимодействия сторонних приложений с GTK 4.0 и/или пользовательскими сервисами.

11.4 Система UNICORE

Проект UNICORE (Uniform Interface to Computing Resources – единый интерфейс к вычислительным ресурсам) зародился в 1997 году, и к настоящему моменту представляет собой комплексное решение, ориентированное на обеспечение прозрачного безопасного доступа к ресурсам грид [65].

Архитектура UNICORE [64] формируется из клиентского, сервисного и системного слоев (см. рис. 59). Верхним слоем в архитектуре является клиентский слой. В нем располагаются различные клиенты, обеспечивающие взаимодействие пользователей с грид средой:

- UCC (Unicore Command Line Client – клиент командной строки для UNICORE): клиент, обеспечивающий интерфейс командной строки для постановки задач и получения результатов;
- URC (Unicore Rich Client – многофункциональный клиент UNICORE): клиент, основанный на базе интерфейса среды Eclipse, предоставляет в графическом виде полный набор всех функциональных возможностей системы UNICORE;
- HiLA (High Level API for Grid Applications – высокоуровневый программный интерфейс для приложений грид): обеспечивает разработку клиентов к системе UNICORE;
- Порталы: доступ пользователей к грид-ресурсам через интернет, посредством интеграции UNICORE и систем интернет-порталов.

Промежуточный сервисный слой содержит все сервисы и компоненты системы UNICORE, основанные на стандартах WSRF и SOAP. Шлюз – это компонент, обеспечивающий доступ к узлу UNICORE посредством аутентификации всех входящих сообщений [48]. Компонент XNJS обеспечивает управление задачами и исполнение ядра UNICORE 6. Регистр сервисов обеспечивает регистрацию и поиск ресурсов, доступных в грид-среде. Также, на уровне сервисного слоя обеспечивается поддержка безопасных соединений, авторизации и аутентификации пользователей.

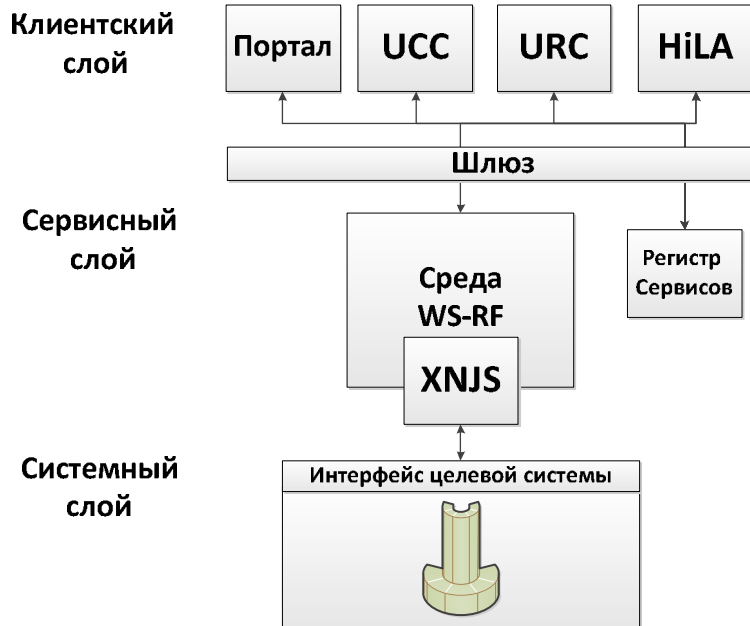


Рис. 59. Архитектура UNICORE 6

В основании архитектуры UNICORE лежит системный слой. Интерфейс целевой системы (TSI – Target System Interface) обеспечивает взаимодействие между UNICORE и отдельным ресурсом грид-сети. Он обеспечивает трансляцию команд, поступающих из грид-среды локальной системе.