

ГРИД И ВЕБ-СЕРВИСНЫЕ ТЕХНОЛОГИИ

*Айтмухамбетов А.А. – д.ф.-м.н., профессор кафедры информатики и математики, Костанайский государственный университет им. А. Байтурсынова
Ким Р.А. – магистрант, Костанайский государственный университет им. А. Байтурсынова*

В данной статье описывается общая концепция грида, классификация и разновидности грид-систем. Исследуются подходы к созданию грид-сервисов, позволяющих создавать простые для проектирования, удобные для реализации и надежные в использовании интерфейсы прикладного программирования, не требующих специальных средств для разработки клиентских программ. Компьютерный грид для обеспечения универсальности обслуживания требует существования единых решений как минимум для решения задач, общих - для любых компьютерных сервисов.

Проведен анализ развития модели ресурсов и протоколов обмена данными, используемых в гридах. Описаны протоколы удаленного вызова процедур, модели ресурсов и протоколы Globus Toolkit первых поколений, открытая архитектура грид-сервисов (OGSA). Особое внимание уделяется стандартам и протоколам, лежащим в основе современных грид-систем: инфраструктуре ресурсов веб-сервисов (Web Service Resource Framework) и семейству спецификаций WS (Web Services). Приведены примеры существующих гридов, проведен анализ используемых в них решений.

Ключевые слова: грид, грид-сервис, веб-сервис, интерфейс, грид-система.

GRID AND WEB SERVICE TECHNOLOGIES

*Aitmukhambetov A.A. - Doctor of Physics and Mathematics, professor of Mathematics and informatics school, A. Baitursynov Kostanay State University
Kim R.A. – the undergraduate, A. Baitursynov Kostanay State University*

This article describes grid's general concept, classification and types of grid systems. It also investigates ways of creating grid services which allow to create applications programming interfaces which are simple to be designed, reliable in usage and handy for implementation and do not require special devices for creation client-side programs. Computer-based grid for providing maintenance generality requires existence of consistent solutions at least for solving problems which are common for all computer-based services.

The analysis of resources model development and communication protocol used in grids was conducted. Protocols of remote procedure call and resource models, first generation Globus Toolkit protocols and open grid service architecture (OGSA) were described. The particular emphasis is given to standards and protocols that form the basis for modern grid systems that are Web Service Resource Framework and Web Services specifications range. The examples of existing grids were given the analysis of solutions used was held.

Key words: grid, grid service, web service, interface, grid system.

ГРИД ЖӘНЕ ВЕБ-СЕРВИСТІК ТЕХНОЛОГИЯСЫ

*Айтмухамбетов А.А. – ф.-м.ғ.д., А. Байтұрсынов атындағы Қостанай мемлекеттік университетінің математика және информатика кафедрасының профессоры
Ким Р.А. – магистрант, А. Байтұрсынов атындағы Қостанай мемлекеттік университеті*

Бұл мақалада гридтың жалпы тұжырымдамасы, грид-жүйесінің әр түрлігі мен жіктелуі сипатталады. Тапсырыскер бағдаламасын жетілдіру үшін арнайы тәсілді қажет етпейтін, іс жүзінде қолданылатын сенімді бағдарламалық интерфейстерді ыңғайлы жүзеге асыру мен қолдану үшін, жобалауда қарапайым нәрселерді құруға мүмкіндік беретін грид-сервисті жасау амалдары зерттелген. Жалпы – кез-келген компьютерлік сервис үшін компьютерлік грид жан-жақты қызмет көрсетуді қамтамасыз ету үшін ең аз дегенде қойылған тапсырманы шешуде біркелкі шешімнің болғанын талап етеді.

Гридта қолданылған мәліметтердің алмасу қорлары мен хаттамалар моделінің дамуына талдау жүргізілді. Шақыру рәсімінің жойылған хаттамалары, қор моделдері мен бірінші буындағы Globus Toolkit хаттамалары, грид-сервистердің (OGSA) ашық құрылымы сипатталған. Осы заманғы грид-жүйесінің негізіне жататын хаттамалар мен стандарттарға ерекше көңіл аударылды: веб-сервис (Web Service Resource Framework) қорының инфрақұрылымы және WS (Web Services) айрықшалау. Қолданыс аясындағы гридтерге мысал келтіріліп, онда қолданылатын қорытындыларға талдау жүргізілді.

Негізгі ұғымдар: грид, грид-сервис, веб-сервис, интерфейс, грид-система.

Грид — это соединение технологии, инфраструктуры и стандартов, позволяющее одним организациям и частным лицам предоставлять ресурсы (компьютеры, хранилища данных, сеть и другие) в общее пользование, а другим — использовать их, когда это необходимо. Технология здесь представляет собой специальное программное обеспечение, которое позволяет обеспечить работу такой системы. Инфраструктура состоит из аппаратных средств и служб (на основе человеческих и программных ресурсов), которые должны быть организованы и поддерживаться для того, чтобы ресурсы могли совместно использоваться. Стандарты должны определять форматы и протоколы обмена сообщениями как между службами, так и между службами и пользователями, а также правила работы грида. Грид отличается от просто большой вычислительной сети, предоставляющей доступ своим пользователям. Существует набор критериев, сформулированных в работе Я. Фостера, по которым можно определить, является ли какая-либо система гридом. Согласно этим критериям, грид

это система, которая:

- координирует использование ресурсов при отсутствии централизованного управления этими ресурсами;

- использует стандартные, открытые, универсальные протоколы и интерфейсы;

- должна обеспечивать высококачественное обслуживание.

Грид строится на базе многоцелевых протоколов и интерфейсов, позволяющих решать такие фундаментальные задачи как аутентификация, авторизация, обнаружение ресурсов и доступ к ресурсам. Если это не так, мы имеем дело со специализированной прикладной системой.

Грид позволяет использовать входящие в его состав ресурсы таким образом, чтобы обеспечивалось высокое качество обслуживания, касающееся, например, таких параметров, как время отклика, пропускная способность, доступность и надежность, а совместное использование ресурсов различных типов, готовых соответствовать сложным запросам пользователей, делает выгоду от использования комбинированной системы значительно выше, чем от суммы ее отдельных частей.

Необходимо отдельно отметить важность стандартных открытых и универсальных протоколов и интерфейсов. Если взять аналогию с электрической сетью, простота и универсальность использования электричества потребителями возможна только благодаря существованию стандартных протоколов и интерфейсов. Интерфейсом можно считать разновидности электрических вилок, обеспечивающих подключение к поставщику электроэнергии, а протоколами — тип, напряжение и частоту электрического тока, подводимого к розеткам. Существуют и более узкие интерфейсы и протоколы, описывающие принципы соединения разных поставщиков услуги энергетических компаний в единую сеть.

Аналогично, компьютерный грид для обеспечения универсальности обслуживания требует существования единых решений как минимум для решения задач, общих для любых компьютерных сервисов.

Среди задач, которые необходимо решить при создании сервиса, предназначенного для работы в среде грид можно выделить следующие:

- задачи обеспечения безопасности (аутентификация и авторизация пользователей, разделение и передача прав);

- задачи обнаружения ресурсов и сервисов;

- задачи выделения и предоставления доступа к ресурсам.

Программные компоненты, направленные на решение этих задач на общем уровне, не специфичном для каких-либо конкретных задач, называют программным обеспечением грид промежуточного уровня (grid middleware).

Первым решением этих задач, ставшим на какое-то время стандартом де-факто для программного обеспечения промежуточного уровня, является Globus Toolkit версии 2.1, разработанный в Аргоннской национальной лаборатории Чикагского университета. Данный проект является именно набором инструментов (toolkit), а не готовым решением; гриды могут использовать его для решения общих задач, перечисленных выше. Среди проектов, использовавших Globus Toolkit первых версий можно назвать EU-DataGrid, GriPhyN, iVDGL.

Опыт практического использования первых версий Globus Toolkit в различных грид-проектах улучшил понимание того, какие общие проблемы должны иметь унифицированное решение во всем гриде. Некоторые компоненты и решения из Globus Toolkit были доработаны и стандартизованы на международном уровне. Например, основная концепция GSI — понятие прокси-сертификатов X.509 и их использование для аутентификации, была доработана и стала стандартом RFC 3820.

Расширения протокола FTP до GridFTP (также известного как GSIFTP), разработанного в рамках GASS стало стандартом де-факто для массовой передачи данных в современных гридах. Кроме того, стала ясна большая роль управления ресурсами в гридах, а также слежения за созданием/выделением ресурсов, и управлением циклом их существования (lifecycle management). Кроме того, стало понятно, что необходимо принятие единых стандартов для улучшения взаимодействия грид-систем друг с другом

Возросший интерес к грид-технологиям со стороны бизнеса и промышленности привел к началу разработки более общих кросс-платформенных стандартов для интерфейсов и протоколов, используемых в грид. Обычные приложения разрабатываются, как правило, для работы на определенной платформе (например, Windows, какая-нибудь разновидность Unix, системы типа mainframe, J2EE и так далее). Возможности, предоставляемые этими платформами, могут очень сильно различаться в плане интегрированных средств управления ресурсами, обеспечения доступа к базам данных, кластеризации, управлению нагрузкой. Однако, не смотря на разнообразие платформ, продолжающееся отделение пользователей компьютерных услуг от их поставщиков делает еще более важной проблему достижения одинакового качества обслуживания в смысле общей семантики безопасности, распределенного управления ресурсами и потоками задач, обработки сбоев, и прочих метрик. В общем случае, грид-технологии направлены на решение именно этих задач, и необходимость открытых стандартов в этой области становится вполне очевидной.

Открытая архитектура грид-сервисов (Open Grid Services Architecture, OGSA), изначально предложенная Я. Фостером, К. Кессельманом и др. в 2002 году, является концепцией для разработки открытых стандартов для грида.

Основной постулат OGSA — все что угодно является *сервисом*: сущностью, имеющей доступ к сети, которая обеспечивает доступ к определенным ресурсам или услугам путем обмена сообщениями. Вычислительные ресурсы, ресурсы хранения данных, сети, программы, базы данных и так далее являются ресурсами, доступ к которым предоставляется посредством сервисов. Принятие такой однородной модели сервисов так же означает, что все компоненты являются виртуальными. Более точно, в OGSA все что угодно представляется грид-сервисом, являющимся Веб-Сервисом, следующим стандартным соглашениям (о контроле доступа, способах вызова и так далее) и поддерживающим стандартные интерфейсы для управления временем жизни сервисов. Этот базисный набор соглашений, которому следуют все без исключения грид-сервисы облегчает создание сервисов более высокого уровня, которые можно рассматривать как более высокий уровень абстракции над базовыми соглашениями.

Существует два понятия, имеющие одно и тоже название: веб-сервис. В данном тексте для избежания путаницы термин «веб-сервис», написанный с маленькой буквы, будет обозначать более широкое понятие: это сервис, использующий для реализации интерфейса прикладного программирования, открытые стандарты веб-технологий: протокол HTTP и другие связанные стандарты (URI, Mime Types и прочие).

Второе более узкое значение будет писаться с большой буквы, «Веб-Сервис»: это сервис, использующий для реализации интерфейса прикладного программирования протокол SOAP поверх HTTP, язык XML для обмена данными и обычно предоставляющий машинно-читаемое описание интерфейса на языке описания Веб-Сервисов (Web Service Description Language, WSDL). В кругах разработчиков приложений, веб-сервисы, описываемые вторым значением этого термина, также известны как «Большие Веб-Сервисы» (Big web services, WS-*2).

Своим появлением веб-сервисы обязаны во многом простоте протокола HTTP, лежащего в основании современной всемирной паутины (World Wide Web). Прототипом веб-сервисов можно считать обычные веб-сайты, содержащие материалы в форматах, хорошо подходящих для машинного разбора (обычно XML). Позже появились библиотеки, реализующие удаленный вызов процедур, использующие протокол HTTP для транспорта, и протоколы типа общего интерфейса шлюза (Common Gateway Interface, CGI) на стороне сервера: XmlRPC, SOAP. Поскольку эти протоколы опирались на удаленный вызов процедур и удаленный вызов объектов в качестве базовых концепций, их реализация сделана по аналогии с другими средствами, реализующими удаленный вызов процедур: интерфейсы описываются на специальном языке описания интерфейсов; на основании этих описаний генерируется программный код (статически или динамически), реализующий сериализацию и передачу соответствующих структур данных.

Такой подход обладает рядом существенных недостатков:

- реализация клиентских и серверных программ во многом зависит от наличия и качества инструментов, работающих с описаниями интерфейсов, реализация программ без использования этих инструментов становится достаточно трудоемкой и чреватой ошибками;

- стандарты, описываемые взаимодействие сервисов достаточно сложны, их реализации в разных библиотеках не всегда полные, и не всегда хорошо совместимые друг с другом;

- не используются многие возможности, заложенные в сам протокол HTTP: кешируемость, идемпотентность и отсутствие побочных эффектов у разных видов запросов;

- адресация сервисов и ресурсов переносится внутрь описаний интерфейсов.

Достоинствами такого подхода является наличие строгих описаний интерфейсов сервисов на языке, предназначенном для автоматического понимания и интерпретации различными программными компонентами. Это позволяет производить автоматическое обнаружение и классификацию сервисов, а в некоторых случаях даже автоматическую стыковку сервисов.

В настоящее время появилась новая концепция реализации веб-сервисов, использующая более полно возможности протокола HTTP для реализации интерфейсов прикладных программ,

основанная на архитектурном стиле передачи состояния представлениями (Representational State Transfer, REST), предложенном Р. Филдингом. Этот стиль также лежит в основе современной редакции протокола HTTP 1.1. Веб-сервисы, построенные с использованием этого архитектурного стиля используют адреса HTTP-ресурсов для адресации ресурсов, с которыми работают, а так же используют методы HTTP для манипуляций с ресурсами и сервисами.

Такие веб-сервисы получили название RESTful-веб-сервисов. Методы создания сервисов с использованием архитектурного стиля REST содержат большую часть «кирпичиков», необходимых для разработки полноценных грид-сервисов в понимании OGSA. Однако, требуется более формальная спецификация следующих ключевых операций:

- создание ресурсов;
- манипуляции со свойствами ресурсов;
- индикация ошибок;
- управление циклом существования ресурсов;
- восстановление после сбоев;
- аутентификация запросов.

Инфраструктура ресурсов веб-сервисов (Web Service Resource Framework, WSRF) появилась как реализация концепций открытой архитектуры грид сервисов при помощи существующих технологий «Больших Веб-Сервисов».

Основной концепцией в архитектуре WSRF является Web Service Resource (WS-Resource). В понимании WSRF, ресурсом является логическая сущность, которую можно отделить от других, возможно аналогичных, сущностей (то есть идентифицировать). Эта сущность может иметь не пустое множество свойств, представимых в виде информационного набора XML (XML infoset). Кроме того, эта сущность может иметь цикл существования: она, возможно, может быть создана по запросу, а так же может иметь ограниченное время жизни. С точки зрения реализации, WS-Resource — это композиция из ресурса и Веб-Сервиса, через который можно получить доступ к этому ресурсу.

Для каждого WS-Resource указывается *конечная ссылка* (End-Point Reference, EPR), указывающая «адрес» ресурса и следующая спецификации WS-Addressing.

Спецификация WSRF так же предусматривает стандартные механизмы для управления циклом существования ресурсов (создание WS-Resource, присвоение ему идентификатора-EPR, уничтожение WS-Resource), получения множества свойств и взаимодействия со свойствами ресурса (с представлением свойств в виде XML infoset).

Инфраструктура ресурсов веб-сервиса разрабатывалась согласно концепциям OGSA, и в данный момент является эталонной реализацией этой архитектуры, вытеснив менее удачную начальную реализацию в виде открытой инфраструктуры грид-сервисов (Open Grid Services Infrastructure, OGSi).

Основными полными реализациями WSRF являются Globus Toolkit версии 4 и UNICORE версии 6.

Поскольку WSRF базируется на больших Веб-Сервисах, она наследует все их недостатки. WSRF фактически расширяет Веб-Сервисы, поэтому зависимость от средств автоматической генерации кода на основе описаний интерфейсов становится еще более заметной, а сложность спецификаций, и как следствие, количество ошибок в реализациях возрастает.

Литература:

1. А.П. Демичев, А.П. Крюков, Л.В. Шамардин. Принципы построения грид с использованием restful-веб-сервисов. Программные продукты и системы, (4): 172-176, 2009.
2. A. Kryukov, A. Demichev, V. Ilyin, L. Shamardin. Architecture of Grid for National Nanotechnology Network (GridNNN). In *Distributed Computing and Grid-Technologies in Science and Education: Proceedings of the 4th Intern. Conf. (Dubna, June 28-July 3, 2010)*, pages 352-356, Dubna, 2010. JINR. 95
3. K. Zyp. A JSON Media Type for Describing the Structure and Meaning of JSON Documents. Technical report, IETF Network Working Group, March 2010. draft-zyp-json-schema-02.
4. А.П. Демичев, В.А. Ильин, А.П. Крюков, Л.В. Шамардин. Реализация программного интерфейса грид-сервиса Pilot на основе архитектурного стиля REST. *Вычислительные методы и программирование*, 11:62-65, 2010.

References:

- 1.A.P. Demichev, A.P. Kryukov, L.V. Shamardin. Printsipy postroyeniya grid s ispolzovaniyem restful - veb-servisov. Programmnye produkty i sistemy, (4): 172-176, 2009.
2. A. Kryukov, A. Demichev, V. Ilyin, L. Shamardin. Architecture of Grid for National Nanotechnology Network (GridNNN). In *Distributed Computing and Grid-Technologies in Science and Education: Proceedings of the 4th Intern. Conf. (Dubna, June 28-July 3, 2010)*, pages 352-356, Dubna, 2010. JINR. 95
3. K. Zyp. A JSON Media Type for Describing the Structure and Meaning of JSON Documents. Technical report, IETF Network Working Group, March 2010. draft-zyp-json-schema-02.

4. A.P. Demichev, V.A. Ilyin, A.P. Kryukov, L.V.Shamardin. Realizatsiya programmnoogo interfeisa grid-servisa Pilot na osnove arkhitekturnogo stilya REST/ Vychislitelnye metody I programmirovaniye, 11: 62-65, 2010.

Сведения об авторах

Айтмухамбетов Абай Ахметкалиевич – доктор физико-математических наук, профессор кафедры информатики и математики, Костанайский государственный университет им. А. Байтурсынова, г. Костанай, ул.Байтурсынова, 47, +7(7142)39-06-28.

Ким Радмила Анатольевна – магистрант 1 курса кафедры информатики и математики, Костанайского государственного университета им. А.Байтурсынова, г. Костанай, ул.Байтурсынова, 47, 8(702)9215996; e-mail: kim_radmila1977@mail.ru.

Aitmukhambetov Abai Akhmetkaliyevich - Doctor of Physics and Mathematics, professor of Mathematics and Informatics school, A. Baitursynov Kostanay State University, Kostanai, 47th Baitursynov street, +7(7142)39-06-28.

Kim Radmila Anatolyevna - The first grade Master's Degree Student of Mathematics and Informatics school of A. Baitursynov Kostanay State University, Kostanai, 47th Baitursynov street, 8(702)9215996, e-mail: kim_radmila1977@mail.ru.

Айтмухамбетов Абай Ахметкалиевич – физика-математика ғылымының докторы, математика және информатика кафедрасының профессоры, А. Байтұрсынов атындағы Қостанай мемлекеттік университеті, Қостанай қ., Байтұрсынов көш., 47, +7 (7142) 39-06-28

Ким Радмила Анатольевна – А. Байтұрсынов атындағы Қостанай мемлекеттік университеті математика және информатика кафедрасының 1 курс магистранты, Қостанай қ., Байтұрсынов көш., 47, 8 (702) 9215996, e-mail: kim_radmila1977@mail.ru.