

# Функциональные возможности и методы реализации программного обеспечения Грид (часть 1)<sup>1</sup>

В.Н. Коваленко

**Аннотация.** Появление распределённых ресурсных инфраструктур, созданных на основе концепции грида, открыло новые возможности для решения важных задач в различных прикладных областях. Однако, решая вопрос об использовании технологий грида, важно учитывать фактическое состояние программного обеспечения в этой сфере. В статье (часть 1,2) предлагается систематизированное изложение функциональных возможностей наиболее распространённых, проверенных временем программных средств для создания грид-инфраструктур и разработки приложений.

**Ключевые слова:** грид, программное обеспечение, Globus Toolkit, gLite, функциональное описание.

## Введение

Направление *грид*, в котором ставится задача создания аппарата для работы с множеством пространственно распределённых ресурсов, получило широкое признание и практическое применение. На основе программных средств грида построено несколько ресурсных инфраструктур, которые интегрируют находящиеся в разных местах компьютерные центры и системы хранения данных, позволяя использовать всю совокупность содержащихся в них ресурсов приложениям из различных областей.

Однако создание *программного обеспечения грида* (ПГО, middleware) не закончено и сейчас переживает важный этап, который можно охарактеризовать как формирование основанной на стандартах, открытой и расширяемой программной платформы, реализующей базовые функции работы с множествами распределённых ресурсов.

В статье [1] мы рассмотрели теоретические положения концепции грида и обосновали воз-

можные способы его применения. В [2] дан анализ принципов организации ПГО, которые представлены в спецификациях Открытой архитектуры грид-служб (Open Grid Services Architecture – OGSA) [3]. Эти два аспекта определяют потенциал грида. Насколько этот потенциал может быть использован на практике, во многом зависит от уровня развития ПГО. Задаче его оценки посвящена данная работа, в которой ставится цель представить фактическое положение дел с реализацией функциональных возможностей грида, имея в виду в качестве эталона архитектуру OGSA.

## 1. Тенденции развития программного обеспечения грида

Дадим, прежде всего, общую картину хода развития ПГО, опираясь на материалы ряда исследований [4-7]. Начало положили в конце 90-х годов всего несколько проектов: Globus, Condor, Legion и Unicore. В них были заложены основы подхода грид, определены основные понятия и

<sup>1</sup> Работа выполнена при поддержке фонда РФФИ (грант № 06-07-89111-а), программы фундаментальных исследований Президиума РАН, гранта Президента РФ для ведущих научных школ НШ-2139.2008.9.

задачи, разработаны базовые технологии. Перспективы, обозначенные стартовыми проектами, получили широкий резонанс, и к 2003-му году только в Европе выполнялись уже 20 крупных проектов (CrossGrid, DAMIEN, DataGrid, DataTAG, GridLab, GRIP, BioGrid, MammoGrid, MOSES, OpenMolGRID и другие), финансируемых Европейским союзом, а также несколько национальных и интернациональных проектов: INFN Grid (Италия), NorduGrid (страны Северной Европы), e-Science (Англия). Эти проекты второй волны существенно опирались на результаты стартовых проектов, но были направлены на решение более широкого круга задач: создание программных средств более высокого уровня, имплементация экспериментальных грид-инфраструктур и разработка приложений.

В проектах второй волны обнаруживаются две тенденции. Во-первых, была осознана важность решения проблемы интероперабельности ПГО, и в нескольких проектах (GEMSS, GRASP, GRIA) использован архитектурный подход, ориентированный на службы. Во-вторых, проекты стали дифференцироваться по разделам программного обеспечения. Если системы, разработанные в стартовых проектах, стремились решать весь круг задач поддержки грида, пользователей и разработчиков, то проекты второй волны получили направленность на отдельные технологии. Как следует из статьи [5], в которой дан обзор положения дел в США и Австралии, эта тенденция характерна не только для европейских проектов. По существу, возникшая дифференциация нашла отражение в современном понимании архитектуры грида OGSA, соответствуя основным функциональным группам служб: интеграции различных типов ресурсов (аппаратных, файловых и структурированных ресурсов данных), обеспечения технологических свойств (безопасность, протоколирование обработки, учёт потребления ресурсов, мониторинг ресурсов). Следует отметить, что тематика проектов не была ограничена только базовым программным обеспечением. От общего числа европейских проектов (двадцать) такие проекты составили половину, 17 проектов относились к разработке приложений, 7 – пользовательских интерфейсов (порталов для доступа в грид).

После 2003 года положение изменилось следующим образом [6]. Большинство европейских проектов сосредоточено на разработке приложений (43 %) и грид-инфраструктур, в то время как разработкой базового ПГО занимаются только 16 %. Эти данные позволяют сделать заключение, что программная база грида оценивается как достаточная для начала его использования на практике. К такому же выводу приходят эксперты консорциума GridCoord [7], считающие, что имеющийся портфель программного обеспечения в целом покрывает базовые технологии грида.

Проблема современного этапа состоит, однако, в том, что реализации технологий выполнены в системах с разной программной архитектурой, что делает невозможным их совместное использование. Принципиальным решением этой проблемы было бы приведение имеющихся программных систем в соответствие со стандартами грида, но многие из них уже задействованы в инфраструктурах, и их трансформация не может быть скорой. По этой причине развитие ПГО вступило в стадию постепенной консолидации программных средств, разработанных в отдельных проектах, и создания на их основе интегрированных программных комплексов.

Реально консолидация осуществляется благодаря появлению крупных проектов (EGEE, Unicore), специализированных институтов (Open Middleware Infrastructure Institute – OMI [8]) и кооперации разработчиков ПГО (Globus Alliance). В результате получено несколько комплексов программного обеспечения, которые свободно распространяются и активно используются для создания гридов и разработки приложений.

В сложившейся ситуации именно крупные комплексы в наибольшей степени отражают возможности современного ПГО, и далее мы даём сопоставительный анализ двух таких комплексов: Globus Toolkit и gLite. Оба эти комплекса получили самое широкое признание, на их основе во многих странах мира созданы производственные гриды. Кроме того, такой выбор является представительным: Globus Toolkit – это единственная эталонная реализация базовой части OGSA, а gLite, во-первых, вобрал в себе результаты разработок многих

предшествующих проектов и, во-вторых, реализует ряд функций, которые служат основой для выработки стандартов.

## 2. Опыт использования программных средств грида

В ИПМ РАН работы в области грида были начаты с момента появления первой версии Globus Toolkit в 1999 г. На базе этого комплекса была образована первая в России распределённая грид-инфраструктура компьютерных ресурсов. Создание такого полигона, в первую очередь, имело целью освоение технологий грида и способов организации ресурсных центров. Результаты работ в этом направлении представлены в публикациях [9-15]. Следующие версии Globus Toolkit были использованы для выполнения различных исследовательских проектов по развитию ПГО, в том числе, по созданию систем диспетчеризации и планирования. Укажем также применение Globus Toolkit в учебных целях: на нём прошло обучение нескольких выпусков студентов семинара “Технологии грида”, действующего на факультете ВМК Московского государственного университета.

Участие ИПМ РАН, в числе других российских институтов, в проекте EGEE (Enabling Grids for ESciencE) позволило внести вклад в создание крупнейшей в мире грид-инфраструктуры, в которой в качестве программного обеспечения используется gLite. Благодаря этому проекту активизировались работы по гриду в России, создан и функционирует в составе общей инфраструктуры российский сегмент RDIG (Russian Data Intensive Grid) [16], число участников которого постоянно увеличивается. ИПМ РАН, внося вклад в организацию работы RDIG в целом, обеспечивает функционирование двух ресурсных центров, один из которых предназначен для предварительного тестирования новых версий программного обеспечения, а на втором выполняется регулярная обработка заданий.

## 3. Комплекс Globus Toolkit

Разработка комплекса Globus Toolkit (GT) началась в конце 90-х годов как совместный проект Аргоннской национальной лаборатории

(ANL) при университете Чикаго и Института информатики университета Южной Каролины (ISI USC). Первоначально проект был направлен на создание программного обеспечения для грида вычислительного типа, который интегрировал глобально распределённые компьютерные центры. Уже в первой версии был предложен принцип включения ресурсов в грид путём установки в ресурсных центрах служб GT и функционально полного набора операций, позволяющих дистанционно работать с компьютерными ресурсами и ресурсами памяти.

Была показана возможность образования глобальной операционной среды, обладающей свойством прозрачности по отношению к административным границам: непосредственно с рабочего места пользователь мог запускать задания, получать результаты их выполнения, хранить и перемещать файлы на любых компьютерах, входящих в состав грида. Функционирование такой среды поддерживалось инфраструктурой безопасности (дистанционные операции выполнялись от имени пользователя и с его правами, которые контролируются при доступе к ресурсам) и информационной службой, осуществляющей динамическое подключение ресурсов и мониторинг их состояния.

Globus Toolkit, начиная с ранних версий, завоевал репутацию стандарта де-факто программного обеспечения грида и применялся как для поддержки грид-инфраструктур, так и в качестве инструментального средства. Среди инсталляций Globus Toolkit указываются [17] гриды: TeraGrid, Open Science Grid, Cancer Bioinformatics Grid, EGEE, LHC Computing Grid, UK National Grid Service, China Grid, China National Grid, NAREGI. С помощью службы GRAM построено несколько диспетчеров заданий: WLMS, Virtual Data System (VDS) [18], Ninf-G [19] и Nimrod-G [20].

Современная версия Globus Toolkit 4 (далее GT4) [17] представляет собой новый крупный шаг в развитии ПГО прежде всего по причине того, что в ней начат переход на архитектуру OGSA.

### 3.1. Исполнительная среда хостинга служб

Следует сказать, что деятельность по стандартизации грида была инициирована и развивается

в большой степени разработчиками Globus Toolkit, а GT4 представляет собой эталонную реализацию ядра OGSA – инфраструктуры, обеспечивающей функционирование служб грида.

Инфраструктура реализована в виде контейнера – среды хостинга, которая поддерживает выполнение установленных в ней Web-служб. В составе GT4 включены 3 контейнера, написанных на языках Java, C и Python. Контейнеры опираются на известные среды хостинга Web-служб с открытым кодом (Tomcat [21], Axis [22]), которые берут на себя ряд функций, осуществляя, например, управление сообщениями и разделение серверных ресурсов между службами. В GT4 реализованы следующие дополнения типовых возможностей контейнеров.

- Главные спецификации Web-служб: WSDL [23], SOAP [24] и WS-Security [25].

- Спецификации управления состоянием: WS-Addressing [26], WS-Resource Framework (WSRF) и WS-Notification [27]. Этим обеспечивается поддержка служб, которые публикуют сведения о собственном состоянии или состоянии обслуживаемых службой ресурсов. Например, в описываемых далее службах GRAM и RFT эти механизмы применяются для управления состоянием ресурсов, ассоциированных с десятками тысяч заданий и процессов передачи файлов.

- Механизм реестра, который содержит информацию о службах, выполняющихся в контейнере и реализованных в соответствии со спецификацией WSRF. На основе этого механизма работают средства мониторинга грида.

- Переход на стандарты OGSA и архитектуру Web-служб сделал Globus Toolkit расширяемым по составу, открыв возможность создания новых служб с использованием имеющихся (что было затруднительно в старых версиях). В то же время для разработки клиентских приложений предусмотрен более простой, по сравнению с инструментальными средствами Web-служб, способ обращения к службам GT4 – прикладной программный интерфейс, представленный в виде клиентских библиотек.

### 3.2. Службы управления заданиями

Служба GRAM (Grid Resource Allocation and Management) реализует схему доступа [1] к компьютерным ресурсам, организованным в

виде кластера и управляемым локальным менеджером. В этой схеме GRAM выполняет роль шлюза, интерпретируя дистанционные операции в среде ресурсных центров.

GRAM является Web-службой, использующей спецификацию WSRF, и предоставляет интерфейсы для мониторинга и управления произвольными вычислениями, которые оформляются в виде заданий, на удалённых компьютерах. Интерфейс запуска задания позволяет клиенту описать:

- тип и количество требуемых для выполнения задания ресурсов;
- данные, которые необходимо передать на исполнительные ресурсы, где будет выполняться обработка;
- исполняемый код и аргументы, используемые при запуске;
- данные безопасности, удостоверяющие владельца задания;
- требования к надежности выполнения.

Группа операций мониторинга даёт возможность опрашивать состояние вычислительных ресурсов и отдельных заданий, подписываться на уведомление об изменении состояния. Для управления заданиями предназначены операции прекращения обработки, модификации параметров и получения результатов.

В стадии разработки находится ещё одна служба – WMS (Workspace Management Service), имеющая отношение к управлению заданиями. Подготавливая выполнение задания, служба GRAM автоматически создаёт для них среду на исполнительном компьютере, однако реализованный в GRAM способ, основанный на использовании статических экаунтов, не решает всех проблем и, в частности, связан с существенными административными затратами по предварительной регистрации пользователей на компьютерах. С помощью имеющейся реализации WMS – Dynamic Accounts клиент может дистанционно создавать на некотором удалённом ресурсе экаунт Unix и управлять им (определять политику доступа и время существования).

В перспективе WMS должна решить и другие задачи виртуализации размещения приложений [28]. Во-первых, многим приложениям требуется вполне определённая среда исполнения: операционная система, библиотеки, архивы данных.

Во-вторых, в интересах как пользователей грида, так и владельцев ресурсов необходимо обеспечить надёжную изоляцию оборудования и программной среды разделяемых ресурсов от выполняющихся на них приложений.

Предлагаемый подход к решению этих проблем основан на концепции виртуального рабочего пространства (VW – virtual workspace), которое по запросу к службе WMS автоматически образуется на ресурсах. Тогда запуск задания происходит за два шага: задание отображается на рабочее пространство, а рабочее пространство отображается на ресурсы грида. VW формально описывается XML-схемой, в которой задаются требования к исполнительной среде: к оборудованию (объём оперативной памяти, диска, сетевые соединения), к программному обеспечению и способу изоляции. В качестве способа реализации VW в ряде проектов (обзор в [29]) ведутся разработки по использованию технологий виртуальных машин.

### 3.3. Средства управления данными

Комплект средств GT4 для работы в данных, представленными в виде файлов, включает: GridFTP, RFT (Reliable File Transfer), RLS (Replica Location Service) и DRS (Data Replication Service).

**Базовые средства** реализованы в соответствии со спецификациями протокола GridFTP [30], предназначенного для надёжного, безопасного и высокопроизводительного перемещения данных между устройствами внешней памяти. Протокол GridFTP представляет собой расширение стандарта FTP для условий грида, в которых требуется:

- дистанционная аутентификация и авторизация без участия пользователя;
- перемещение больших объёмов данных (порядка тера- и петабайтов) между системами памяти;
- доступ к большим объёмам данных (порядка гига- и терабайтов) из пространственно распределённых приложений.

Реализация GridFTP в GT4 преследует цель обеспечить дистанционный доступ к несовместимым между собой системам управления внешней памятью. Делается это обычным для грида способом – путём введения промежуточ-

ного сервера-шлюза, который преобразует унифицированные запросы GridFTP в протоколы конкретных систем управления. Шлюз абстрагирует стоящий за ним источник или приёмник данных и за счёт сменных модулей способен обеспечивать доступ к обычным файловым системам, а также к системам хранения, подобным HPSS [31] и SRB [32].

В состав реализации протокола GridFTP входят сервер, пользовательский интерфейс командной строки и библиотеки для разработки приложений. С помощью средств разработки приложения могут осуществлять доступ к удалённым файлам в соответствии со стандартом POSIX, то есть с помощью операций read/write/open/close.

GridFTP обладает следующими особенностями.

- *Аутентификация, целостность и конфиденциальность данных.* Поддерживается аутентификация управляющего канала (RFC 2228) и канала данных, а также управляемые пользователем уровни целостности и конфиденциальности.

- *Управление передачей данных третьей стороной* (Рис. 1). Такого рода управление даёт возможность приложению, находящемуся в некоторой точке, инициировать, контролировать и управлять передачей данных между двумя другими точками: источником и получателем.

Надёжность и высокая производительность достигается с помощью ряда механизмов.

- *Параллельная передача.* Может быть организована передача между источником и потребителем несколькими TCP-потоками, что

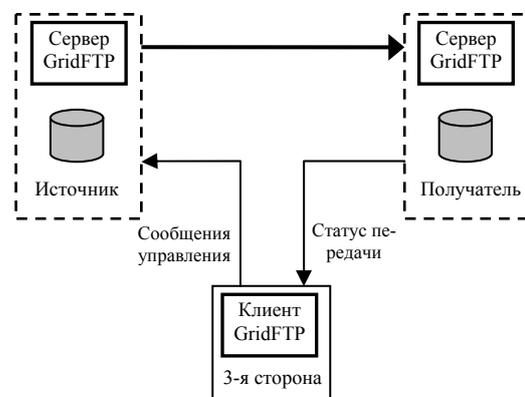


Рис. 1. Управление передачей данных третьей стороной

способствует увеличению суммарной полосы пропускания.

- *Расслоённая (striped) передача.* В параллельных системах памяти данные из одного файла могут храниться на нескольких серверах, и GridFTP способен поддерживать передачу таких файлов.

- *Частичная передача файла.* Существует возможность передавать произвольную часть файла.

- *Подбор размеров буфера и окна TCP.* Известно, что параметры буфера и окна TCP оказывают существенное влияние на производительность передачи. GridFTP имеет набор команд для ручной и автоматической установки этих параметров (в GT4 реализована только ручная установка).

- *Поддержка надёжной рестартуемой передачи.* Этот механизм позволяет бороться с временным выходом из строя сети или сервера путём возобновления передачи с нужной точки.

**Служба RFT.** Хотя протокол GridFTP используется во многих проектах грида и стал стандартом де-факто, он не лишён недостатков. В частности, способ вызова операций передачи файла в нём принципиально отличается от интерфейсов Web-служб. Весьма существенно и то обстоятельство, что надёжность GridFTP ограничена требованием, чтобы клиент сохранял жизнеспособность в течение всего времени передачи. Если передача длится часами, клиентская машина должна быть постоянно включена, а это во многих случаях затруднительно, например, для мобильных клиентов.

По этим причинам на основе GridFTP разработана служба Reliable File Transfer (RFT) [33]. Интерфейс этой службы “надёжной передачи” аналогичен интерфейсу службы GRAM управления вычислениями. Клиент может запустить “задание” по передаче данных (потенциально миллионов файлов) и быть свободным, возлагая на RFT его выполнение. RFT представляет собой службу WSRF, так что состояние передачи сохраняется в базе данных и восстанавливается в случае падения службы или выключения компьютера. Пользователь может получить состояние задания по передаче или подписаться на получение оперативного уведомления. Отметим, что RFT используется в качестве сред-

ства доставки файлов в службе управления заданиями GRAM.

**Службы репликации.** Развитие программных средств более высокого уровня, настроенных над GridFTP и RFT, представлено в GT4 службами Replica Location Service (RLS) [34] и Data Replication Service (DRS) [35].

Служба обнаружения RLS предназначена для решения одной из проблем репликации, а именно, проблемы определения физического расположения реплик файлов по их логическим именам. Логическое имя файла (LFN) – это уникальный идентификатор, используемый пользователями для обозначения файла. Функция службы обнаружения заключается в том, чтобы найти все физические копии (реплики), соответствующие заданному LFN. Каждая физическая копия определяется физическим именем файла (PFN) (для GridFTP это URL), которое задаёт расположение реплики в некоторой системе хранения.

С учётом требований приложений архитектура RLS исходит из необходимости поддерживать десятки миллионов файлов, сотни миллионов реплик, сотни обновлений в секунду, тысячи систем хранения. Эти требования выдвигают на первый план вопросы масштабируемости и надёжности, которые решаются с помощью многоуровневой пространственно распределённой архитектуры. Первый уровень представлен локальными каталогами реплик (LRC), каждый из которых содержит информацию о репликах одной системы хранения. Функция LRC – поддержка соответствия между логическими и физическими именами файлов, хранящимися в этой системе.

Совокупность LRC содержит всю информацию о существующих репликах, но запрос к любому из них не может найти все реплики. Поэтому вводятся второй и последующие уровни, занимаемые индексами RLI (Replica Location Indices) (Рис. 2). Информация, хранящаяся в индексе – это пары (LFN, указатель на LRC), которые заносятся и обновляются в соответствии с информацией, поступающей от LRC.

Каждый элемент распределённого каталога (LRC или RLI) периодически посылает одному или нескольким элементам вышележащего уровня информацию о своём состоянии, ис-

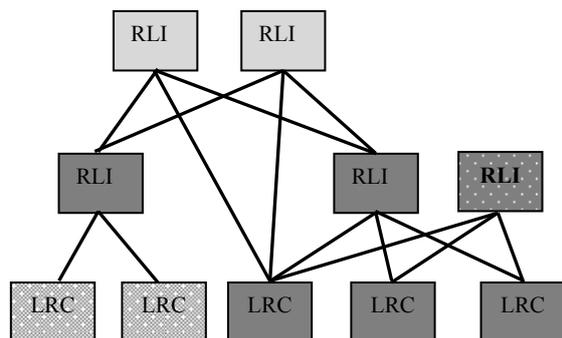


Рис. 2. Иерархическая организация RLS

пользуя различные варианты алгоритма распространения, и таким образом могут быть построены разнообразные структуры индексов. В алгоритме распространения состояния для ускорения поиска используется разбиение пространств логических имён файлов и физических имён реплик, а для обеспечения надежности – избыточность в индексном пространстве.

Служба DRS (Data Replication Service) реализует функцию создания реплик, опираясь на RLS и GridFTP. Получая список файлов, DRS проверяет наличие их реплик в обслуживаемой системе хранения, используя для этого локальный каталог реплик, и передаёт в эту систему копии отсутствующих файлов из других систем хранения, регистрируя их в каталоге. DRS реализована как Web-служба, использующая WSRF, однако в GT4 имеет статус предварительной версии.

### 3.4. Службы информационного обслуживания

В составе GT4 имеются две компоненты, относящиеся к тематике информационных задач грида – MDS4 [36, 37] и OGSA-DAI. Система OGSA-DAI (Data Access and Integration), разработанная в программе UK e-Science, реализует общий подход к интеграции информации на основе архитектуры OGSA. Здесь мы остановимся на службе обнаружения и мониторинга MDS4 (Monitoring and Discovery System).

Важность этой компоненты определяется динамичным характером такой среды как грид: непосредственно в ходе функционирования в неё могут добавляться новые ресурсы и службы, а старые могут изыматься или становиться временно недоступными. Осуществляемый

MDS4 оперативный сбор информации об их состоянии позволяет решать две задачи:

- обнаружение ресурсов и служб, входящих в состав грида;
- мониторинге их состояний и детектирование проблемных ситуаций.

MDS4, как и аналогичные ей средства, существенно используется в других компонентах программного обеспечения грида – диспетчерах заданий и службах обеспечения надежного выполнения.

Реализация MDS в GT4 отличается от предыдущих версий тем, что она основана на стандартных спецификациях WSRF и WS-Notification, которые задают правила публикации свойств (в формате XML) элементов грида (служб и ресурсов, моделируемых службами) с возможностью получения значений этих свойств путём опроса или подписки на уведомления. В отличие, например, от реестра UDDI [38], в MDS4 могут публиковаться сложно структурированные данные большого объёма. Хранимые данные имеют определённое время жизни, по окончании которого они автоматически удаляются.

В состав MDS4 входят две службы: Index и Trigger, собирающие данные от различных источников, в качестве которых могут выступать:

- службы, имеющие интерфейсы WSRF и WS-Notification;
- произвольные программы, генерирующие данные;
- другие системы мониторинга.

Служба Index предоставляет клиентам WSRF-интерфейсы типа запрос и подписка на поставку хранящихся в ней данных. Одна индексная служба может регистрироваться в другой такой службе, чтобы автоматически получать от неё данные. С помощью механизма регистрации осуществляется многоуровневое агрегирование данных от разных индексных служб. При установке MDS4 в контейнере создаётся экземпляр службы Index, в котором автоматически регистрируются все другие использующие WSRF службы GT4 (GRAM, RFT и т.д.).

Служба Trigger также собирает данные, но в ней выполняются конфигурируемые действия в соответствии с задаваемыми условиями на значения данных. Пример такого действия – по-

ссылка e-mail сообщения при исчерпании дискового пространства на ресурсе памяти.

### 3.5. Службы обеспечения безопасности

Аппарат безопасности GT4 решает следующие задачи:

- защита целостности и конфиденциальности сообщений;
- аутентификация – установление идентичности пользователя или службы;
- авторизация – определение допустимости выполнения тех или иных действий.

Соответствующие функции реализованы непосредственно в контейнере, и, таким образом, сама среда хостинга обеспечивает безопасность всех установленных в ней служб. Кроме того, отдельно поддерживается ряд функций по управлению данными безопасности (credentials) и информацией о виртуальных организациях.

Средства безопасности основываются на стандарте X.509 документов безопасности – сертификатов и временных сертификатов. Первые применяются для однозначной идентификации пользователей и серверов, вторые – для временного делегирования их полномочий действующим от их лица программным объектам.

Реализация базовых компонентов безопасности выполнена в соответствии со стандартами Web-служб и включает следующие механизмы.

- Механизм безопасности уровня сообщений реализует стандарты WS-Security и WS-SecureConversation, обеспечивая защиту сообщений SOAP.
- Безопасность транспортного уровня и аутентификация обеспечиваются протоколом TLS [39].
- Делегирование реализовано с помощью временных сертификатов X.509 на основе спецификации WS-Trust.
- Оболочка безопасности позволяет создавать разнообразные схемы авторизации: список контроля доступа (ACL) на основе файла отображения в локальную среду “grid-mapfile”; ACL, определяемый внешней службой; частный программный блок авторизации; получение решений по авторизации от внешней службы по протоколу SAML [40].

Дополнительные средства безопасности в составе GT4 представлены системами хранения

сертификатов и получения их по запросу MyProxy [41] и Community Authorization Service (CAS) [42].

Служба CAS поддерживает функционирование виртуальных организаций, осуществляя хранение информации о её членах и информации о политике доступа к ресурсам. Поддерживается следующий сценарий деятельности: для доступа к ресурсам грида пользователь контактирует со службой CAS, которая делегирует права пользователю, исходя из запрашиваемого им действия и выполняемой им роли в виртуальной организации (ВО). При выполнении операции на ресурсе полученные права предъявляются оболочке безопасности ресурса, которая выносит решение по авторизации путём сравнения прав пользователя в ВО и прав, делегированных ВО владельцем ресурса.

### Литература

1. В.Н. Коваленко, Д.А. Корягин. Базовые принципы и способы применения грида. Программирование, № 1, 2009, стр. 26-49.
2. В.Н. Коваленко, Е.И. Коваленко, Л.В. Ухов. Принципы стандартизации программного обеспечения грид. Информационные технологии и вычислительные системы, Москва, РАН, 2009, № 2, стр. 81 – 95.
3. I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich. The Open Grid Services Architecture, Version 1.5. Global Grid Forum, July, 2006.
4. Paul Graham, Matti Heikkurinen, Jarek Nabrzyski, Ariel Oleksiak, Mark Parsons, Heinz Stockinger, Kurt Stockinger, Maciej Stroinski, Jan Weglar. EU Funded Grid Development in Europe. 2nd European Across Grids Conference, Nicosia, Cyprus, January 2004, Springer Verlag.
5. Mark Baker, Rajkumar Buyya, Domenico Laforenza, Grids and Grid Technologies for Wide-Area Distributed Computing, Software: Practice and Experience (SPE), Volume 32, Issue 15, Pages: 1437-1466, Wiley Press, USA, December 2002.  
<http://www.gridbus.org/papers/gridtech.pdf>
6. Grids research in Europe. An overview prepared by GridCoord. Luxembourg: Office for Official Publications of the European Communities, 2006.  
<http://www.gridcoord.org/grid/portal/information/public/reports/GridCoord-Grid%20Research%20In%20Europe.doc>
7. Grid R&D Vision and Roadmap, Summary Report. The GridCoord Consortium: September 2006.  
[http://www.gridcoord.org/grid/portal/information/public/reports/Vision\\_and\\_RM\\_V.2.0\\_220906.pdf](http://www.gridcoord.org/grid/portal/information/public/reports/Vision_and_RM_V.2.0_220906.pdf)
8. Open Middleware Infrastructure Institute – OMII. <http://www.omii.ac.uk/>
9. Коваленко В.Н., Корягин Д.А. Вычислительная инфраструктура будущего. Открытые системы, № 11-12

- (1999), стр. 45-52, <http://www.osp.ru/os/1999/11-12/045.htm>
10. Коваленко В.Н. Проблемы сетевых файловых систем. Открытые системы, №3 (1999), стр. 9-15, <http://www.osp.ru/os/1999/03/03.htm>
  11. Коваленко В.Н., Коваленко Е.И. Пакетная обработка заданий в компьютерных сетях. Открытые системы, № 7-8 (2000), стр. 1-19. <http://www.osp.ru/os/2000/07-08/010.htm>
  12. Волков Д.В. Кластерная система Condor. Открытые системы, № 7-8 (2000), стр. 20-26. <http://www.osp.ru/os/2000/07-08/020.htm>
  13. В.Н. Коваленко, Е.И. Коваленко, Д.А. Корягин, Э.З. Любимский, Е.В. Хухлаев. Современное состояние и направления развития программного обеспечения GRID. Информационные технологии и вычислительные системы, № 4, Москва, РАН, 2003, стр. 23- 36.
  14. Коваленко В.Н., Корягин Д.А. Оценка возможностей программных платформ Грид. Труды международной конференции «Распределенные вычисления и Грид-технологии в науке и образовании» Дубна, 29 июня - 2 июля 2004 г., стр. 128-132.
  15. В.Коваленко, Д.Корягин. Организация грид: есть ли альтернативы? Открытые системы, № 12 (2004)
  16. Российский Грид для интенсивных операций с данными (РДИГ/EGEE). <http://www.egee-rdig.ru/>
  17. Ian Foster. Globus Toolkit Version 4: Software for Service-Oriented Systems. H. Jin, D. Reed, and W. Jiang (Eds.): NPC 2005, LNCS 3779, pp. 2 – 13, 2005. IFIP International Federation for Information Processing, 2005.
  18. Deelman E., Singh G., Su M.H. et al. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. Scientific Programming, 2005, 13(3): 219-237.
  19. Tanaka Y., Nakada H., Sekiguchi S et al. Ninf-G: A reference implementation of RPC based programming middleware for grid computing. Journal of Grid Computing, 2002, 1(1): 41-51.
  20. Abramson D., Giddy J., Kotler L. High performance parametric modeling with Nimrod/G: Killer application for the global grid? In Proc. The Int. Parallel and Distributed Processing Symposium (IPDPS), 2000, pp.520-528.
  21. Apache Tomcat. <http://tomcat.apache.org/>
  22. Apache Axis. <http://ws.apache.org/axis/>
  23. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Recommendation, June 2007. <http://www.w3.org/TR/wsdl20/>
  24. Simple Object Access Protocol (SOAP). <http://www.w3.org/TR/soap/>
  25. OASIS Web Services Security (WSS). <http://www.oasis-open.org/committees/wss/>
  26. Web Services Addressing (WS-Addressing). W3C Member Submission 10 August 2004. <http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/>
  27. K. Czajkowski, D. F. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, W. Vambenepe. The WS-Resource Framework. March 5, 2004. <http://www.globus.org/wsrif/specs/ws-wsrf.pdf>
  28. Keahey K, Foster I, Freeman T, Zhang X. Virtual workspaces: Achieving quality of service and quality of life in the grid. Scientific Programming, 2005, 13(4): 265-275.
  29. Ильин В.А., Крюков А.П., Шамардин Л.В., Демичев А.П., Горбунов И.Н. Способ запуска и обработки в гриде заданий, подготовленных для различных сред исполнения. Вычислительные методы и программирование. Том 9, 2008, с. 41-47, [http://num-meth.srcc.msu.su/zhurnal/tom\\_2008/v9r206.html](http://num-meth.srcc.msu.su/zhurnal/tom_2008/v9r206.html)
  30. GridFTP Protocol Specification (Global Grid Forum Recommendation GFD.20). W. Allcock, editor. March 2003.
  31. High Performance Storage System (HPSS). <http://www.hpss-collaboration.org/>
  32. SDSC Storage Resource Broker (SRB). <http://www.sdsc.edu/srb>
  33. Reliable Data Transport: A Critical Service for the Grid. W.E. Allcock, I. Foster, R. Madduri. Building Service Based Grids Workshop, Global Grid Forum 11, June 2004.
  34. Giggie: A Framework for Constructing Scalable Replica Location Services. A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunst, M. Ripeanu, B. Schwartzkopf, H. Stockinger, K. Stockinger, B. Tierney. Proceedings of Supercomputing 2002 (SC2002), November 2002.
  35. Wide Area Data Replication for Scientific Collaborations. A. Chervenak, R. Schuler, C. Kesselman, S. Koranda, B. Moe. Proceedings of 6th IEEE/ACM International Workshop on Grid Computing (Grid2005), November 2005.
  36. Jennifer M. Schopf, Laura Pearlman, Neill Miller, Carl Kesselman, Ian Foster, Mike D'Arcy, Ann Chervenak. Monitoring the grid with the Globus Toolkit MDS4. Journal of Physics: Conference Series 46 (2006), 521–525, SciDAC 2006.
  37. A Globus Primer. [http://www.globus.org/toolkit/docs/4.0/key/GT4\\_Primer\\_0.6.pdf](http://www.globus.org/toolkit/docs/4.0/key/GT4_Primer_0.6.pdf)
  38. "UDDI Standard", <http://www.uddi.org>, 2006.
  39. RFC 4346. The Transport Layer Security (TLS) Protocol. Version 1.1. <http://tools.ietf.org/html/rfc4346>.
  40. Security Assertion Markup Language (SAML). <http://xml.coverpages.org/saml.html>
  41. An Online Credential Repository for the Grid: MyProxy. J. Novotny, S. Tuecke, V. Welch. Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.
  42. A Community Authorization Service for Group Collaboration. L. Pearlman, V. Welch, I. Foster, C. Kesselman, S. Tuecke. Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.

**Коваленко Виктор Николаевич.** Заведующий сектором Института прикладной математики им. М.В. Келдыша РАН. Окончил Московский физико-технический институт в 1973 г. Кандидат физико-математических наук. Автор более 60 научных работ. Область научных интересов: распределенные системы, технологии грида. E-mail: [kvn@keldysh.ru](mailto:kvn@keldysh.ru).