

на правах рукописи

Шамардин Лев Витальевич

Принципы построения грида с использованием RESTful-веб-сервисов

Специальность 05.13.11 — математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей.

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата физико-математических наук

Москва 2011

Работа выполнена в отделе теоретической физики высоких энергий
Научно-исследовательского института Ядерной Физики имени
Д.В. Скобельцина Московского Государственного Университета имени
М.В. Ломоносова

Научный руководитель:

кандидат физико-математических наук Крюков Александр Павлович.

Официальные оппоненты:

доктор технических наук, профессор
Бухановский Александр Валерьевич,
Санкт-Петербургский Государственный Университет Информационных
Технологий, Механики и Оптики (СПбГУ ИТМО);

кандидат физико-математических наук
Жуматий Сергей Анатольевич,
Научно-исследовательский вычислительный центр Московского Государ-
ственного Университета имени М.В. Ломоносова (НИВЦ МГУ).

Ведущая организация:

Учреждение Российской академии наук
Институт прикладной математики им. М.В. Келдыша РАН

Защита диссертации состоится «13» мая 2011 года в 15 часов на заседании
диссертационного совета Д 501.002.09 Московского государственного универ-
ситета имени М.В. Ломоносова по адресу 119991, г. Москва, Ленинские горы,
д. 1, стр. 4, НИВЦ МГУ, конференц-зал.

С диссертацией можно ознакомиться в библиотеке НИВЦ МГУ.

Автореферат разослан «11» апреля 2011 года

Ученый секретарь

диссертационного совета



Суворов В.В.

Общая характеристика работы

Актуальность работы

В настоящее время все большее распространение приобретает использование суперкомпьютеров и вычислительных кластеров для расчетов, связанных с решением задач моделирования в различных научных областях: исследование свойств новых материалов, моделирование погодных явлений, исследования процессов термоядерного синтеза и формирования галактик, моделирование в областях биофизики и биохимии, поиск новых лекарств, анализ экспериментальных данных в физике элементарных частиц. Массовое использование технологий высокопроизводительной обработки данных радикально сокращает затраты на реализацию самых передовых и самых сложных проектов. Суперкомпьютеры позволяют ускорить расчеты, делая возможным практическое применение более точных и всесторонних математических моделей. Развитие суперкомпьютерных технологий и организация их эффективного использования является одним из приоритетных научных направлений Российской Федерации.

Проведение расчетов на суперкомпьютерах связано с решением большого количества технических и организационных задач, связанных с обеспечением доступа к ресурсам и повышением эффективности их использования. В настоящее время получили развитие концепции грида и грид-вычислений, направленные на обеспечение удаленного доступа к географически распределенным вычислительным ресурсам и услугам, поэтому представляется перспективным их использование для организации удаленной работы на суперкомпьютерах. С помощью грида от конечного пользователя скрываются подробности внутреннего устройства вычислительных ресурсов, предоставляя ему унифицированный способ доступа к этим ресурсам. Существующие в настоящее время стандарты и методы в области грид-вычислений на основе спецификации WSRF используют достаточно сложные в реализации программные интерфейсы веб-сервисов, требующие, как правило, специальных библиотек или средств автоматической генерации программного кода. В последнее время активно развиваются альтернативные подходы к созданию веб-сервисов, основанные на архитектурном стиле REST («RESTful-веб-сервисы»). Разработка методов создания грид-сервисов на основе таких подходов позволяет упростить интерфейсы грид-сервисов, тем самым расширяя возможности их прямого использования прикладными программами.

Важными проблемами, связанными с использованием гридов, являются распределение выполняемых задач между узлами сети, а также автоматизация сложных вычислений, состоящих из отдельных независимых шагов, являющихся, обычно, самостоятельными вычислительными задачами. Для их выполнения используются грид-сервисы, обеспечивающие «оркестровку»,

то есть последовательный или одновременный запуск отдельных шагов композитной задачи в соответствии с заданной логикой и отслеживание процесса их выполнения. Такие грид-сервисы выполняют центральную роль во многих существующих грид-проектах. Разработка грид-сервиса, обеспечивающего запуск задач, и имеющего простой и надежный программный интерфейс, построенный на основе новых методов, базирующихся на архитектурном стиле REST является важной прикладной задачей и хорошей проверкой практической ценности и эффективности предложенных методов.

Цель и задачи исследования

Основными целями данной диссертационной работы являются:

- разработать подход к созданию грид-сервисов, позволяющий создавать простые для проектирования, удобные для реализации и надежные в использовании интерфейсы прикладного программирования, не требующие специальных средств для разработки клиентских программ;
- реализовать грид-сервис запуска композитных заданий, имеющий удобный интерфейс прикладного программирования, построенный на разработанном подходе, обеспечивающий эффективное управление потоками заданий в гриде Национальной Нанотехнологической Сети (ГридННС);
- провести практические испытания разработанных программных средств на полигоне грид-инфраструктуры.

Положения, выносимые на защиту

1. Разработан новый подход к созданию грид-сервисов с использованием архитектурного стиля REST, который позволяет создавать простые для реализации интерфейсы прикладного программирования, не требующие использования специальных средств для разработки клиентских и серверных программ. Важными свойством нового подхода является универсальность форматов представлений ресурсов, упрощенная модель восстановления при сбоях и концептуальная простота. Впервые рассмотрен вопрос управления циклом существования ресурсов для сервисов такого типа.
2. Реализован грид-сервис запуска композитных заданий Pilot, имеющий интерфейс прикладного программирования, следующий разработанному подходу, предназначенный для управления потоками композитных заданий, имеющих структуру направленных ациклических графов.

3. Проведены тестовые испытания грид-сервиса запуска композитных заданий Pilot в рамках полигона ГридННС, подтверждающие правильность предложенного подхода. В процессе опытной эксплуатации на полигоне ГридННС грид-сервис Pilot продемонстрировал высокую производительность. Грид-сервис Pilot является законченным программным продуктом, используемым в настоящее время на полигоне ГридННС.

Научная новизна

В диссертации впервые сформулированы методы разработки грид-сервисов с использованием архитектурного стиля REST, в том числе универсальные методы управления временем жизни ресурсов RESTful-веб-сервисов. Обобщены существующие лучшие методики и на их основе даны строгие формулировки принципов построения грид-сервисов с использованием RESTful-веб-сервисов. Предложенный подход позволяет обеспечить универсальность форматов представлений ресурсов, упрощенную модель восстановления при сбоях и концептуальную простоту реализации. Таким образом, становится возможным создание удобных и надежных интерфейсов прикладного программирования, не требующих использования специальных средств или библиотек как для разработки клиентских программ так и самих грид-сервисов.

Практическая значимость работы

В работе сформулированы методы, применяемые при разработке нового грид сервиса, обеспечивающие концептуальную простоту его интерфейса прикладного программирования.

На основе предложенных методов разработан новый грид-сервис запуска композитных заданий, названный «Pilot», который используется в существующем гриде национальной нанотехнологической сети для запуска композитных заданий и контроля за их выполнением. Реализованный грид-сервис является модульным решением, которое позволяет применять его не только в инфраструктуре ГридННС, но и в других гридах и e-инфраструктурах, поддерживающих совместимые интерфейсы грид-шлюзов суперкомпьютеров. Кроме того, имеющаяся реализация может быть легко расширена для использования альтернативных интерфейсов грид-шлюзов, что обеспечивает ее интеграцию в другие гриды.

Разработка данного сервиса была выполнена в связи с потребностью в надежном грид-сервисе запуска композитных заданий, который имел бы достаточно удобный для использования интерфейс прикладного программирования, не привязанный к конкретным средствам разработки программного обеспечения или библиотекам. Работоспособность предложенных решений проверялась на практике на полигоне проекта ГридННС. Реализованный в

рамках работы грид-сервис прошел испытания и используется в инфраструктуре проекта ГридННС.

Личный вклад автора

Личный вклад автора состоит в формулировке предлагаемых в диссертации оригинальных методов построения грид-сервисов, разработке форматов описания композитных заданий и задач, разработке интерфейса прикладного программирования грид-сервиса Pilot и последующей реализации грид-сервиса Pilot.

Основное содержание диссертационной работы и ее результатов полностью отражено в четырех научных публикациях автора. В материалах совместных публикаций, за исключением публикации [4], личный вклад автора является определяющим. В публикации [4] личный вклад автора заключается в разработке технологических требований к регистрационным центрам инфраструктуры публичных ключей, используемой для надежной аутентификации запросов к грид-сервисам, а также практической проверке всех требований, приведенных в разделе 3 публикации.

Апробация работы

Работа обсуждалась и получила положительную оценку на следующих научных семинарах:

- на научном семинаре отделения Программирования Института Прикладной Математики имени М.В. Келдыша Российской Академии Наук (16 февраля 2010 г.);
- на научном семинаре Parallel.ru Научно-Исследовательского Вычислительного Центра Московского Государственного Университета имени М.В. Ломоносова (15 марта 2010 г.);
- на научном семинаре «Проблемы современных информационно-вычислительных систем» Механико-Математического Факультета Московского Государственного Университета имени М.В. Ломоносова под руководством д.ф.-м.н., проф. В. А. Васенина (16 ноября 2010 г.).

Основные результаты работы, разработанный подход к построению грид-сервисов, описание реализации грид-сервиса Pilot были доложены автором на следующих международных научных конференциях:

- на Четвертой международной конференции «Распределенные вычисления и Грид-технологии в науке и образовании» (GRID'2010) (28 июня–3 июля 2010 г., Дубна);

- на симпозиуме летней школы XtreamOS 2010 (XtreamOS Summer School 2010, doctoral symposium) (5–9 июля 2010 г., Гюнзбург), доклад был отмечен дипломом «best presentation award»;
- на 18-й Международной конференции по вычислениям в физике высоких энергий и ядерной физике (18th International Conference on Computing in High Energy and Nuclear Physics, CHEP 2010) (18–22 октября 2010 г., Тайпей).

Структура и объем работы

Диссертация состоит из введения, пяти основных глав, заключения, библиографии и двух приложений. Объем диссертации составляет 132 страницы включая приложения, объем библиографии — 70 ссылок, работа содержит 14 рисунков и 6 таблиц.

Краткое содержание работы

Первая глава, «Грид и веб-сервисные технологии», описывает общую концепцию грида, классификацию и разновидности грид-систем. Проведен анализ развития модели ресурсов и протоколов обмена данными, используемых в гридах. Рассматривается современная модель гридов на основе открытой архитектуры грид-сервисов (Open Grid Services Architecture, OGSA). Указаны основные проблемы в текущих подходах к реализации OGSA на основе инфраструктуры ресурсов веб-сервисов (Web Service Resource Framework, WSRF): сложность стандартов, плохая совместимость различных реализаций, зависимость от специальных средств генерации кода или библиотек, недоступных для многих языков программирования. Дается общее представление об архитектурном стиле REST и применении его в области веб-сервисов, рассказывается о RESTful-веб-сервисах. Ставится задача реализации грид-сервисов, концептуально близких OGSA, на основе этого архитектурного стиля.

Далее осуществляется краткий обзор существующих в настоящее время грид-инфраструктур, предназначенных для обработки данных физических экспериментов и для высокопроизводительных вычислений. Показывается ключевая роль сервиса, отвечающего за распределение задач между вычислительными узлами грида. Сравниваются характеристики распространенных систем распределения задач в гриде. Ставится задача реализации грид-сервиса запуска композитных заданий в гриде на основе архитектурного стиля REST.

Во второй главе, «Использование REST для построения грид-сервисов», описаны методы и подходы к созданию грид-сервисов в рамках

архитектурного стиля «передачи состояния представлениями» (Representational State Transfer, REST). Суть архитектурного стиля REST заключается в следующем. Ресурс R — это изменяющаяся во времени функция членства $M_R(t)$, которая в момент времени t отображается на множество сущностей или значений, эквивалентных в этот момент времени. Значения в этом множестве могут быть представлениями ресурса, и/или идентификаторами ресурса (они используются для адресации ресурсов). Ресурсы могут быть как статическими, так и динамическими во времени. Единственное свойство ресурса, которое никогда не должно меняться — это семантика самого отображения, поскольку именно она отличает ресурсы друг от друга. Сторона, присваивающая идентификатор конкретному ресурсу в рамках отведенного ей пространства имен, принимает на себя обязательство поддерживать семантическую правильность отображения с течением времени (не изменять функцию членства).

Архитектурный стиль REST подразумевает использование модели взаимодействия клиент-сервер. В применении к гриду, сервер обеспечивает доступ к ресурсу, имеющему свое определенное состояние и время жизни (с точки зрения OGSA такая совокупность из ресурса и сервера является грид-сервисом). Коммуникации между клиентом и сервером, тем не менее, не имеют собственного состояния, то есть каждый запрос клиента содержит в себе всю информацию, необходимую для его понимания, независимо от того, какие запросы поступали до него. Таким образом забота о поддержании состояния коммуникаций является полностью ответственностью клиента, а не сервера.

В настоящее время в различных источниках опубликованы методы, описывающие использование архитектурного стиля REST для создания веб-сервисов. Такие веб-сервисы называют RESTful-веб-сервисами. В работе автором сформулированы более строгие определения и требования к RESTful-веб-сервисам, которые приводятся ниже.

Ресурс — это сущность, обладающая достаточной информационной ценностью, чтобы ссылка на нее имела практический смысл.

Унифицированный идентификатор ресурса (Uniform Resource Identifier, URI) — это строка, определяющая имя и адрес ресурса или его представления. Существует также важная частная форма URI, называемая *унифицированное имя ресурса* (Uniform Resource Name, URN). Унифицированное имя ресурса используется для адресации ресурсов, для которых не предполагается обязательной доступности по сети. При разработке RESTful-веб-сервисов следует выбирать унифицированные идентификаторы, обладающие описательными свойствами, то есть обеспечивающие интуитивное соответствие идентификатора адресуемому ресурсу.

Представление ресурса — это последовательность битов в каком-либо формате, отражающая сущность ресурса. Для ресурсов, представляющих собой реальные физические объекты либо сущности, которые не могут быть

редуцированы до чистой информации, в качестве представлений выбираются документы, содержащие полезную для потребителя сервиса информацию о состоянии ресурса.

Связи между ресурсами — это информация о каких-либо отношениях между ресурсами, возможно доступными через разные сервисы. Связи содержат полные или частичные URI ресурсов, имеющих отношение к данному, а также указания на суть отношения между ресурсами.

RESTful-веб-сервис должен обладать следующими четырьмя ключевыми свойствами:

- *Адресуемость*: любые (все возможные) аспекты набора данных, доступных сервису, являются ресурсами. Поскольку каждому ресурсу соответствует по крайней мере один URI, это означает, что грид-сервис предоставляет URI для любого элементарного фрагмента информации, доступ к которому он может обеспечить.
- *Отсутствия состояния сервиса*: каждый запрос к грид-сервису рассматривается в полной изоляции от других запросов. Вся информация, необходимая для интерпретации и понимания запроса, содержится в самом запросе; информация, следующая из контекста предыдущих запросов к сервису, никогда не используется. Это не означает, что ресурс, доступ к которому обеспечивается сервисом, не имеет состояния. Это означает, что сервис «не помнит», к какому из ресурсов направляется данный запрос клиента, эта информация должна передаваться явно в самом запросе. Однако сам ресурс при этом имеет вполне определенное состояние, которое может изменяться в результате выполнения запросов, посылаемых клиентом или внешних факторов.
- *Связности*: все представления ресурсов сервиса, кроме, возможно, представлений, идентификаторы которых генерируются по алгоритмическим правилам, описанным в спецификации интерфейса прикладного программирования, образуют односвязный направленный граф. Свойство связности является очень важным для RESTful-веб-сервисов, так как информация о связях между ресурсами может использоваться клиентами для обнаружения идентификаторов интересующих ресурсов.
- *Однородности интерфейса*: для выполнения любых операций с ресурсами используются стандартные методы протокола HTTP. При этом выполняемая операция должна концептуально совпадать с используемым методом.

Данные определения понятий, связанных с RESTful-веб-сервисами и их свойств являются обобщением методик, ранее опубликованных другими авторами.

Далее в работе более подробно рассматривается использование протокола HTTP для организации однородного интерфейса грид-сервисов. Доступ к свойствам и представлениям ресурсов осуществляется при помощи метода GET протокола HTTP. Для создания ресурсов используются HTTP-запросы POST по адресу коллекции ресурсов, для изменения ресурсов — запросы PUT по адресу ресурса. При этом допускается интерпретация запросов PUT по адресу существующего ресурса, содержащих лишь частичное представление ресурса, как запросов на изменение только части свойств, содержащихся в представлении. В силу свойства адресуемости отдельные параметры ресурса могут быть доступны как самостоятельные ресурсы. В диссертации проведено сравнение запросов на изменение единичных параметров таких ресурсов для WSRF-сервисов, и RESTful-веб-сервисов, и показано, что использование последних сокращает объем пересылаемой служебной информации более чем на 80% по сравнению с WSRF. Для простых параметров, таких как числа или короткие строки, это уменьшает суммарный размер запроса более чем на 70%.

В работе предложен оригинальный подход к формированию запросов по выбору части свойств ресурса, либо подмножества ресурсов коллекции, при помощи компонентов `query` и `fragment` в URI для указания запроса и используемого языка (например: `https://example.com/carts/1234?//[local-name()="ProductCode"]#XQuery` для запроса только свойств ресурса с именем `ProductCode`, используется язык запросов `XQuery`).

Показана эквивалентность базовых классов ошибок WSRF некоторым стандартным кодам ошибок HTTP, а также приводится интерпретация и рекомендации по использованию прочих кодов ошибок HTTP при реализации RESTful-веб-сервисов

Впервые рассматривается вопрос управления циклом существования ресурсов для RESTful-веб-сервисов. Предложен новый заголовок `Termination-Time`, назначение которого — индикация гарантированного времени существования ресурса (в ответе сервиса) а также дополнительное указание на необходимость изменения времени жизни ресурса (в запросе к сервису). Наличие заголовка `Termination-Time` в ответе грид-сервиса указывает, что данный ресурс может быть автоматически уничтожен сервисом в указанный момент времени, однако не гарантирует уничтожения ресурса. Отсутствие заголовка `Termination-Time` в ответе грид-сервиса означает, что время жизни ресурса не определено явно, и клиент не должен делать предположений о времени жизни ресурса.

Для изменения времени жизни ресурса без совершения каких-либо операций с ресурсом предлагается использовать расширенное указание `Pragma: only-termination-time`. Чтобы изменить время жизни ресурса, не изменяя ресурс, клиент отправляет серверу запрос PUT, содержащий заголовок `Pragma: only-termination-time` и желаемое значение времени жизни ре-

сурса в заголовке `Termination-Time`, но без представления ресурса. Запрос, содержащий представление ресурса и такой набор заголовков считается ошибочным.

Рассмотрен подход к восстановлению ошибок связи с сервисом, использующий свойства отсутствия побочных эффектов и идемпотентности некоторых методов протокола HTTP. Данный подход не имеет аналогов в WSRF, поскольку в последнем используется метод `POST`, не являющийся ни безопасным, ни идемпотентным. Обсуждаются способы обеспечения идемпотентности при создании новых ресурсов, основанные на использовании расширенной `Post Once Exactly`, либо генерации идентификаторов ресурсов клиентом и комбинации условных запросов `PUT` и заголовка `Expect` протокола HTTP/1.1. С учетом отсутствия состояния у коммуникаций между клиентом и сервисом, использование только безопасных или идемпотентных методов при обращении к сервису позволяет существенно упростить процесс восстановления после ошибок связи: в случае отсутствия ответа сервиса клиенту достаточно в точности повторить запрос, ответ на который не был получен.

Отдельное внимание уделяется вопросам аутентификации запросов к веб-сервисам. Анализируются стандартные схемы аутентификации HTTP, `Basic` и `Digest`, а также схема `Atom Authentication`. Указываются особенности этих схем, делающих их практическое применение в гриде неприемлемым. Рассматриваются возможности аутентификации запросов при помощи использования HTTP внутри TLS-соединений (HTTPS) с обязательной верификацией сторон при помощи сертификатов X.509. Приводятся основные методы создания инфраструктуры публичных ключей, позволяющие обеспечить надежную аутентификацию с использованием сертификатов X.509. Данные методы были разработаны в рамках работ по проекту EU-DataGrid с участием автора.

В конце главы подводится краткий итог методам и подходам, рекомендуемым авторам для создания RESTful-веб-сервисов, предназначенных для грида.

Третья глава, «Разработка грид-сервиса запуска композитных заданий», начинается с общего описания грида национальной нанотехнологической сети (ГридННС) и характерных расчетов, производимых пользователями этого грида. Формулируется определение композитного задания, как ациклического графа, узлы которого являются отдельными элементарными исполняемыми *задачами*, а ребра задают зависимость задач друг от друга, таким образом указывая порядок выполнения отдельных задач. Вводится понятие требований к ресурсам и программному окружению задачи. Дается метод описания заданий и задач как JSON-документов. Приводятся формальные схемы, определяющие синтаксис описаний. Такой выбор представления для заданий, задач и требований к ресурсам понятен человеку для непосредственного чтения и записи, не содержит чрезмерной избыточности

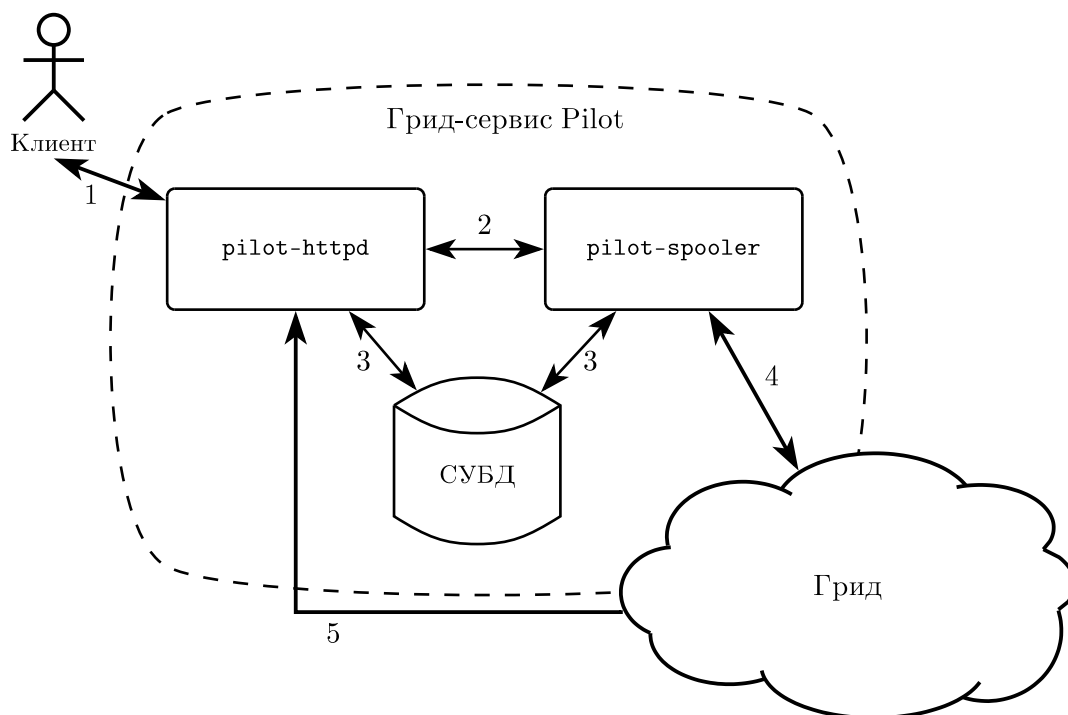


Рис. 1: Общая структура грид-сервиса Pilot (1 — запросы клиентов сервиса, 2 — внутренние связи сервисов `pilot-httpd` и `pilot-spooler`, 3 — обращения к базе данных, 4 — запросы на запуск задач на грид-шлюзах, опрос состояния задач, 5 — уведомления о ходе выполнения задач).

и удобен для разбора или генерации с использованием практически любых современных языков программирования.

Формулируется набор функций, выполняемых грид-сервисом запуска композитных заданий Pilot. На основе методов, описанных во второй главе, строится интерфейс прикладного программирования грид-сервиса Pilot. Приводятся основные ресурсы и коллекции ресурсов сервиса, описывается способ работы с операциями, выполнение которых требует длительного времени: такие операции реализуются как самостоятельные ресурсы, относящиеся к заданиям, к которым их необходимо применить. Сервис обрабатывает такие операции по мере возможности и изменяет свойства ресурса-операции, указывая на завершение процесса обработки. Описывается модель авторизации доступа к ресурсам грид-сервиса.

В четвертой главе, «Реализация грид-сервиса Pilot», описаны подробности реализации грид-сервиса Pilot. Сервис разделен на две независимые программы-демона, выполняющие определенные наборы функций, написанные на языке Python. Для хранения данных используется внешняя СУБД (поддерживаются PostgreSQL и SQLite). Общая структура сервиса приведена на рисунке 1.

Программа внешнего интерфейса `pilot-httpd` отвечает за обработку всех внешних запросов к грид-сервису Pilot. Он обрабатывает запросы клиентов

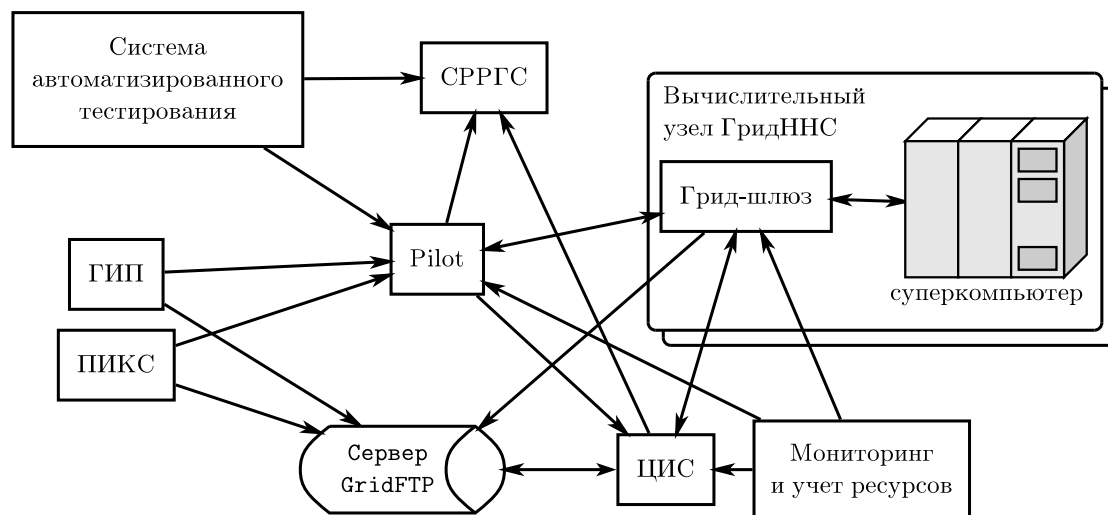


Рис. 2: Схема взаимодействий компонентов полигона ГридННС. Направления стрелок от инициатора запросов (клиента) к соответствующему сервису. (СРРГС — сервис регистрации ресурсов и грид-служб, ГИП — Графический Интерфейс Пользователя, ПИКС — пользовательский интерфейс командной строки, ЦИС — центральная информационная система).

сервиса, обеспечивает аутентификацию и авторизацию пользователей. Фактически, именно здесь содержится реализация серверной части интерфейса прикладного программирования сервиса. Обработка определенных запросов требует обращения к внутреннему интерфейсу сервиса *pilot-spooler*, а также обращения к базе данных. Кроме того, эта программа принимает уведомления о ходе выполнения задач от грид-шлюзов.

Программа обработки задач *pilot-spooler* обеспечивает выполнение логики обработки заданий, запускает задачи на грид-шлюзах и следит за ходом их выполнения. Информация о заданиях и задачах берется из базы данных. Кроме того, *pilot-spooler* обращается к информационным системам грида и грид-шлюзов для получения информации о состоянии и конфигурации грид-ресурсов, а также реализует подбор и сортировку ресурсов, удовлетворяющих требованиям задач.

Пятая глава, «Использование грид-сервиса Pilot», описывает практическое использование и испытания сервиса на полигоне ГридННС. К полигону ГридННС подключено 11 вычислительных кластеров и суперкомпьютеров из различных научных институтов. Перечислены основные компоненты полигона, взаимодействующие с грид-сервисом Pilot, их назначение.

Рассказывается о взаимодействии грид-сервиса Pilot с другими компонентами полигона ГридННС (рисунок 2), в том числе с центральной информационной системой (ЦИС), сервисом регистрации ресурсов и грид-служб (СРРГС). Приводятся критерии рассмотрения грид-сайтов в качестве кандидатов на запуск задач сервисом. Описывается автоматизированная система тестирования полигона, запускающая тестовые задания через грид-сервис

	DagMan	gLite WMS	UNICORE	Taverna	Pilot
поддержка композитных заданий	+	+	+	+	+
абстрактные композитные задания	+	+	–	+	+
режим работы в виде грид-сервиса	\pm^1	+	+	+	+
передача данных между задачами	–	M, P^2	M^2	C^2	C^2
возможность запуска параллельных задач	+	+	+	\pm	+
подбор ресурсов по параметрам	–	+	–	\pm	+
подбор ресурсов по предустановленному ПО	–	\pm^3	–	–	+
поддержка частичного выполнения заданий	–	–	+	+	+
групповые зависимости между задачами	–	–	–	–	+
подключение кластеров через шлюзы	+	–	+	+	+

Примечания:

¹ Работа в виде грид-сервиса, возможно, при использовании внешнего грид-шлюза.

² Режимы передачи данных: M — использование внешнего сервиса, управляющего репликацией данных; C — централизованная передача через один сервис; P — передача от узла к узлу

³ gLite WMS поддерживает подбор ресурсов по тегам, которые соответствуют предустановленному ПО.

Таблица 1: Сравнение функциональных характеристик популярных систем запуска заданий, поддерживающих композитные задания, с грид-сервисом Pilot.

Pilot на все зарегистрированные вычислительные ресурсы. Сервис показал устойчивую работу на полигоне ГридННС, в настоящее время с его помощью запускается в среднем более 1000 заданий в сутки.

Далее проводится сравнение функциональных возможностей сервиса с другими популярными сервисами распределения задач в гриде: DagMan (в реализациях Condor и GridWay), gLite WMS, UNICORE, Taverna. По сравнению с существующими аналогами (таблица 1) сервис Pilot поддерживает более тонкую настройку выбора подходящих ресурсов: подбор по версиям предустановленного программного обеспечения, групповые зависимости — возможность запуска нескольких шагов на одном и том же, не известном заранее ресурсе. Сервис Pilot поддерживает упрощенную централизованную модель передачи данных.

В конце главы приводятся результаты сравнительного тестирования скорости запуска задач через Pilot и грид-шлюз ГридННС по сравнению с запуском через gLite WMS и вычислительный элемент gLite CREAM CE. Тесты показали, что расходы времени, связанные с собственной работой грид-среды, для запуска через Pilot и ГридННС оказались в несколько раз меньше, чем при использовании ПО gLite.

В заключении приведены основные результаты диссертационной работы, выносимые на защиту:

- Разработан новый подход к созданию грид-сервисов с использованием архитектурного стиля REST, который позволяет создавать простые для реализации интерфейсы прикладного программирования, не требующие использования специальных средств для разработки клиентских и серверных программ. Важным свойством нового подхода является универсальность форматов представлений ресурсов, упрощенная модель восстановления при сбоях и концептуальная простота. Впервые рассмотрен вопрос управления циклом существования ресурсов для сервисов такого типа.
- Реализован грид-сервис запуска композитных заданий Pilot, имеющий интерфейс прикладного программирования, следующий разработанному подходу, предназначенный для управления потоками композитных заданий, имеющих структуру направленных ациклических графов.
- Проведены тестовые испытания грид-сервиса запуска композитных заданий Pilot в рамках полигона ГридННС, подтверждающие правильность предложенного подхода. В процессе опытной эксплуатации на полигоне ГридННС грид-сервис Pilot продемонстрировал высокую производительность. Грид-сервис Pilot является законченным программным продуктом, используемым в настоящее время на полигоне ГридННС.

Приводятся данные о публикациях автора по теме диссертации и об апробации работы.

Публикации по теме диссертации

Публикации в журналах из перечня ВАК:

1. Демичев А.П., Ильин В.А., Крюков А.П., Шамардин Л.В. *Грид-технологии на службе компьютеринга для Большого адронного коллайдера* // Информатизация образования и науки — 2009 — №4. — М.: ФГУ ГНИ ИТТ «Информика», — С. 158–171.
2. Демичев А.П., Крюков А.П., Шамардин Л.В. *Принципы построения грид с использованием restful-веб-сервисов* // Программные продукты и системы — 2009 — №4. — г. Тверь, — С. 172–176.
3. Демичев А.П., Ильин В.А., Крюков А.П., Шамардин Л.В. *Реализация программного интерфейса грид-сервиса Pilot на основе архитектурного стиля REST* // Вычислительные методы и программирование — 2010 — Том 11 — С. 65–65.

4. J. Aсталос, R. Cecchini, B. Coghlan, R. Cowles, U. Epting, T. Genovese, J. Gomes, D. Groep, M. Gug, A. Hanushevsky, M. Helm, J. Jensen, C. Kanellopoulos, D. Kelsey, R. Marco, I. Neilson, S. Nicoud, D. O'Callaghan, D. Quesnel, I. Schaener, L. Shamardin, D. Skow, M. Sova, A. Waananen, P. Wolniewicz, W. Xing. *Peer Review and Policy Management Through the European DataGrid Certification Authority Coordination Group // Lecture Notes in Computer Science (Advances in Grid Computing — EGC 2005) — 2005 — vol. 3470/2005 — Springer Berlin / Heidelberg — pp. 285–295.*

Публикации в других научных изданиях:

1. A. Kryukov, A. Demichev, V. Ilyin, L. Shamardin. *Architecture of Grid for National Nanotechnology Network (GridNNN)*. Distributed Computing and Grid-Technologies in Science and Education: Proceedings of the 4th Intern. Conf. (Dubna, June 28–July 3, 2010). — Dubna: JINR, Д-11-2010-140, 2010. — pp. 352–356. / Распределенные вычисления и Грид-технологии в науке и образовании: Труды 4-й междунар. конф. (Дубна, 28 июня–3 июля 2010 г.) — Дубна: ОИЯИ, Д-11-2010-140, 2010. — С. 352–356.
2. L. Shamardin, A. Demichev, A. Kryukov, V. Ilyin. *GridNNN Job Execution Service: a RESTful Grid Service*. Distributed Computing and Grid-Technologies in Science and Education: Proceedings of the 4th Intern. Conf. (Dubna, June 28–July 3, 2010). — Dubna: JINR, Д-11-2010-140, 2010. — pp. 215–219. / Распределенные вычисления и Грид-технологии в науке и образовании: Труды 4-й междунар. конф. (Дубна, 28 июня–3 июля 2010 г.) — Дубна: ОИЯИ, Д-11-2010-140, 2010. — С. 215–219.