

*Технологические основы
грид-инфраструктуры для
Национально
нанотехнологической сети*

А.Крюков, В.А.Ильин*

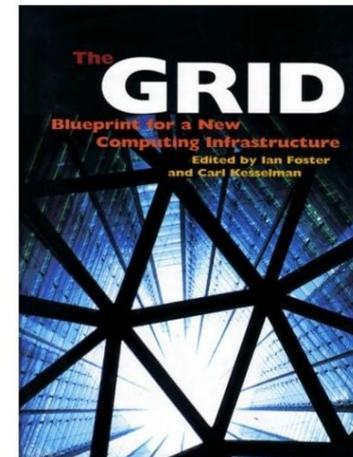
НИИЯФ МГУ

**kryukov@theory.sinp.msu.ru*

Что такое грид?

Ян Фостер. Грид – это система, которая:

- ▶ координирует использование ресурсов при отсутствии централизованного управления этими ресурсами
- ▶ использует стандартные, открытые, универсальные протоколы и интерфейсы
 - аутентификация, авторизация,
 - обнаружение ресурсов, доступ к ресурсам.
- ▶ обеспечивать высококачественное обслуживание
 - время отклика, пропускная способность,
 - доступность и надежность.



http://gridclub.ru/library/publication.2004-11-29.5830756248/publ_file/

Что такое грид?

- ▶ Грид – географически распределенная инфраструктура.
 - Объединяет ресурсы разных типов (процессоры, долговременная и оперативная память, хранилища и базы данных, сети).
 - Доступ к ним пользователь может получить из любой точки, независимо от места их расположения.
- ▶ Грид предполагает коллективный разделяемый режим доступа к ресурсам и к связанным с ними услугам в рамках глобально распределенных виртуальных организаций.
 - В каждой виртуальной организации имеется своя собственная политика поведения ее участников, которые должны соблюдать установленные правила.

HPC – High Performance Computing

➤ Суперкомпьютерные вычисления

- Системы чрезвычайно малой латентности.
 - Параллельные вычисления с частым обменом небольшими порциями данных между процессорами.
 - Специализированные технологии межпроцессорной связи.
 - Системы общей памяти. **Дорого!**

▶ Кластерные вычисления

- Системы среднего уровня латентности.
 - Параллельные вычисления, массовые вычисления.
 - Серийные решения межпроцессорной связи (Infiniband,...).
 - Средний и малый уровень цены.

▶ Грид вычисления

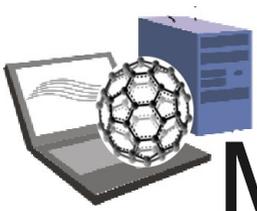
- Системы высокой латентности.
 - Массовые вычисления.
 - Глобальные (Интернет) инфраструктуры.
 - Географически разделенные ресурсы.

Виртуальные организации

- ▶ Все пользователи грида распределены по виртуальным организациям (ВО).
- ▶ ВО – это абстракция для обозначения группы пользователей, институтов и ресурсов, принадлежащих единой администрируемой области.
- ▶ Например, в WLCG ВО соответствуют реальным организациям или проектам:
 - LHC экспериментам,
 - Коллаборации в области биоинформатики,
 - Международный проект по управляемому термоядерному синтезу,
 - И т.д.

Грид для национальной нанотехнологической сети

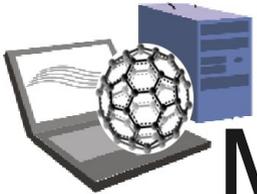
- ▶ ГридННС – это проект, который развивается в рамках целевой программы «Национальная нанотехнологическая сеть» (ННС).
- ▶ Проект стартовал в конце 2008г.
- ▶ Цель проекта:
 - Обеспечить доступ участников ННС к распределенным высокопроизводительным вычислительным ресурсам, которые имеются и планируется создать в рамках ННС.



Модель прикладной области

- **Большое количество независимых коллективов (ВО) — сотни.** Так как нанотехнологические исследования, как следует уже из самого названия, очень близко к технологии, то есть к коммерческому применению, каждая группа исследователей будет достаточно изолирована от другой группы. Вопросы разделения прав доступа предполагается решать на основе технологии ВО.
- **Небольшое количество членов в одной ВО — десятки человек.** Данная оценка исходит из того, что в этой области для решения проблем не требуется образования многотысячных коллективов. Основной упор делается на многообразии групп, а не на многообразии исследовательских тем в одной большой группе.

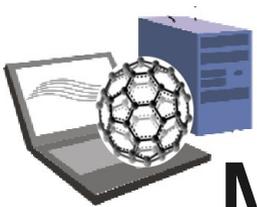




Модель прикладной области

- **Небольшое количество ресурсных центров — десятки.** Основным локальным ресурсом, который предполагается предоставлять пользователям — это вычислительные ресурсы на базе небольших и средних суперкомпьютерных центров. Представляется, что число таких центров не будет превышать нескольких десятков.



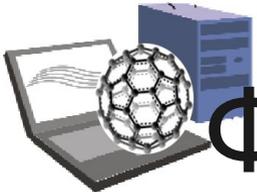


Модель прикладной области

- Небольшой объем входных/выходных данных — гигабайты.
- Большой объем параллельных вычислений (суперкомпьютеры)
 - десятки и сотни ЦПУ,
- время вычислений — дни и возможно недели.

Как правило, процесс обработки задания состоит из множества этапов и для достижения конечной цели необходимо правильно организовать исполнения этапов. Это приводит к необходимости поддержки сложных, композитных заданий.

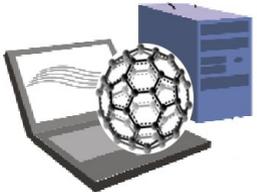




Функциональные требования

- Универсальный и прозрачный доступ к ресурсам
- Возможность использования специализированных веб-сервисов
- Управление потоками заданий ("оркестровка" веб-сервисов)
- Возможность запуска параллельных заданий
- Пакетный режим выполнения заданий
- Управление пользователями и виртуальными организациями
- Мониторинг
- Учет использования ресурсов
- Централизация и децентрализация служб (подсистем) грида.





Облачные вычисления

vs. ГридННС

- Основной потребитель — это массовый пользователь с небольшими (стандартными) запросами к предоставляемым ресурсам.

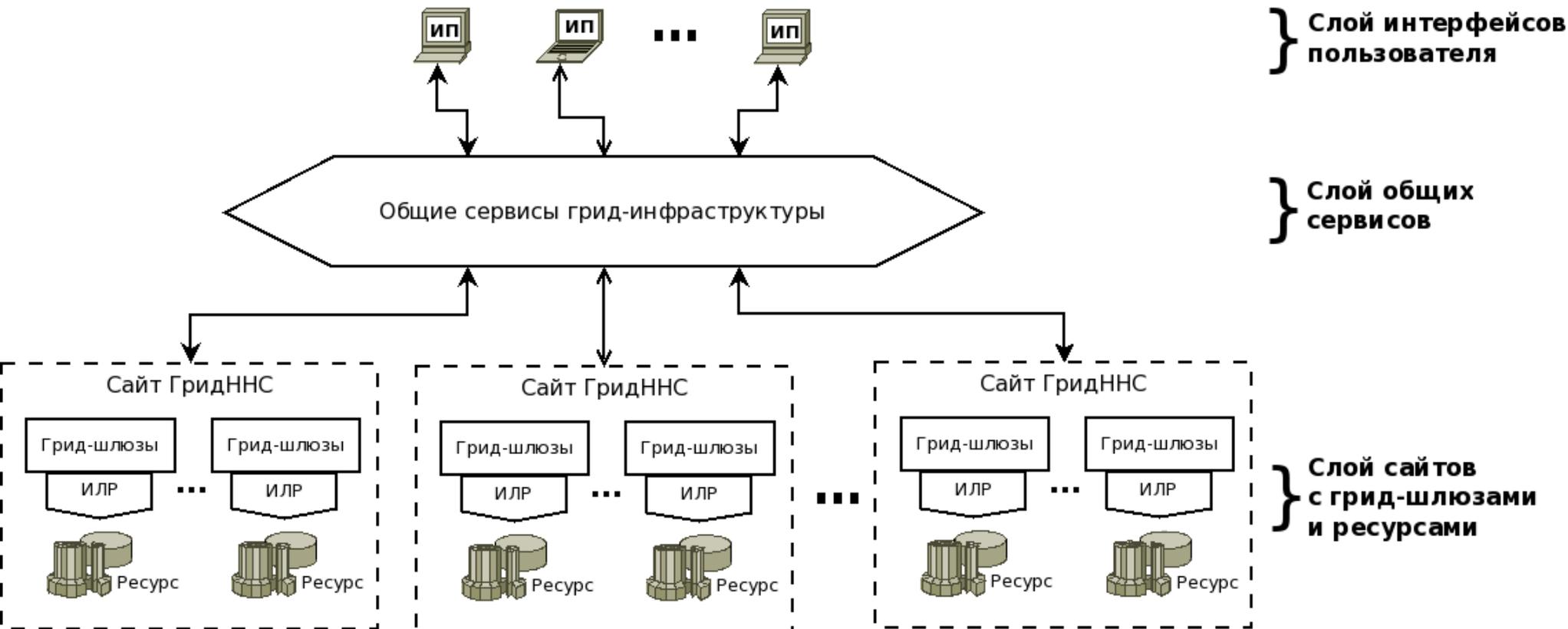
- Основной потребитель — это небольшая исследовательская группа, которым нужен доступ к суперкомпьютеру



Используемое ППО в ГридННС

- ▶ Globus toolkit-4
 - WS-GRAM
 - Gridftp
 - WS-MDS
- ▶ Ряд компонентов из других проектов
 - VOMS (gLite)
- ▶ Оригинальные разработки
 - Система управления заданиями — Пилот
 - Система мониторинга
 - Система регистрации сервисов
 - ...

Структура ГридННС



Грид-шлюзы

- Слой грид-шлюзов обеспечивает доступ к компьютерным ресурсам из ГридННС.
 - В настоящее время ими являются вычислительные ресурсы и ресурсы хранения данных, представляющие собой GridFTP серверы.
 - В качестве грид-шлюза к вычислительным ресурсам используются WS-GRAM из GT4 с некоторыми расширениями.
 - В частности расширен набор поддерживаемых менеджеров локальных ресурсов (МЛР). Разработаны адаптеры для МЛР Cleo [9] и Slurm.

Общесистемные сервисы ГридННС

- Слой общесистемных сервисов и систем отвечает за функционирование и управление ГридННС.
Перечислим основные из них:
 - информационный сервис;
 - система мониторинга и учета;
 - сервис управления и распределения задач;
 - система регистрации сервисов и ресурсов;
 - служба выдачи и управления цифровыми сертификатами;
 - сервис проверки работоспособности ресурсов.

Принципы разработки промежуточного программного обеспечения ГридННС

- При построении ГридННС широко учитывался опыт работы с другими грид-инфраструктурами полученный авторами и, в первую очередь, опыт проектирования, развертывания и эксплуатации ресурсных центров и региональных базовых сервисов WLCG/EGEE.
- WLCG/EGEE является крупнейшей в мире, вышедшей на производственный уровень. Тем не менее, в процессе ее проектирования был принят ряд, по мнению авторов, не очень удачных решений, которые мы постарались улучшить.



Архитектурный стиль REST и RESTful сервисы в гриде

WSRF сервисы

RESTful сервисы

взаимодействие с ресурсами

использование представлений ресурсов

использование одного сервиса для работы с множеством однородных ресурсов

WSRF

- ▶ Веб-сервис
- ▶ XML для представления данных
- ▶ SOAP для обмена сообщениями (или вызова методов)
- ▶ Спецификации WS-* чтобы формализовать понятия ресурса и другие связанные концепции
- ▶ HTTP(S) в качестве транспортного протокола

WSRF-сервисы: ресурсы

Ресурс (WS-Resource)

Сущность, обладающая следующими свойствами:

- может быть идентифицирован;
- может иметь непустое множество свойств;
- может иметь цикл существования.

- ▶ WSRF: Web Services Resource Framework
- ▶ Идентификация:
- ▶ WS-Addressing, End-Point
- ▶ Reference (EPR)

WSRF-сервисы: использование SOAP

- ▶ SOAP рассматривает HTTP только как транспортный протокол.
 - Адрес сервиса - единственный URL.
 - Идентификация ресурсов на основе дополнительных параметров внутри SOAP, возможности адресации HTTP не используются.
- ▶ Сообщения SOAP используют сложные и громоздкие XML-конструкции.
- ▶ Генерация таких сообщений вручную сложна.
- ▶ Обязательное использование каких-либо формальных языков описания сервисов.
- ▶ Реализация как сервисов так и клиентов фактически требует генерации программного кода на основе описаний сервисов, либо использования специальных библиотек

Абстракция REST: веб сервисы

- ▶ Протокол HTTP сам построен в рамках абстракции REST.
- ▶ Возможно реализовать веб сервисы, непосредственно использующие протокол HTTP для описания вызовов – RESTful веб сервисы.
- ▶ Каждый ресурс имеет свой идентификатор - URI.
- ▶ Однородные по типу ресурсы объединяются в коллекции.
- ▶ Принципиально соответствие между выполняемой логической операцией и используемым HTTP-методом.

RESTful веб сервисы: HTTP-методы и ресурсы

- ▶ Пример адреса (URI) ресурса:
`http://example.com/resources/123`
- ▶ GET: получение представления ресурса.
- ▶ PUT: изменение ресурса или создание нового ресурса с заранее определенным идентификатором.
- ▶ POST: создание нового дочернего ресурса, то есть ресурс рассматривается как коллекция.
- ▶ DELETE: удаление ресурса.

RESTful веб сервисы в гриде

- ▶ RESTful и WSRF сервисы имеют много общего:
 - взаимодействие с ресурсами
 - использование представлений ресурсов
 - использование одного сервиса для работы с множеством однородных ресурсов

RESTful-грид-сервисы

- ▶ Возможность управления циклом существования
- ▶ Использование свойства идиempотентности операций
- ▶ Единообразная обработка ошибок
- ▶ Использование протокола HTTP(S) не только как транспортный протокол.

Управление циклом жизни Grid-сервиса

- Управление циклом существования ресурсов является одной из ключевых особенностей Grid-сервисов, не свойственной обычным веб-сервисам.
 - В Gridе постоянно происходит создание и уничтожение ресурсов
 - Необходим единый механизм для информирования о времени жизни сервисов/ресурсов, а так же предсказуемого изменения этого времени жизни.
- Нестандартный тег **Termination-Time** - указывает, что данный ресурс может быть автоматически уничтожен сервисом в данный момент времени, однако не гарантирует уничтожения ресурса.

Свойство идемпотентности REST запросов

- Действия, выполняемые методами GET и HEAD не должны выполнять никаких функций помимо собственно возвращения представления ресурсов и не должны вызывать изменения внутреннего состояния сервиса или ресурса.
- Действия, выполняемые методами PUT и DELETE должны быть идемпотентными, то есть побочные эффекты, вызываемые запросом с таким методом должны быть одинаковы для любого количества повторений одного и того же запроса.
- Создание ресурса (POST) — единственный не идемпотентный метод.

Обработка ошибок

- RESTful-грид-сервис использует для индикации ошибок стандартные коды статусов HTTP со значениями 4xx и 5xx.
- В качестве базового класса используется код состояния 400 Bad Request, которое указывается для любых ошибок, не попадающие в один из стандартных классов, перечисленных далее.
- Обращения к неизвестному ресурсу - код состояния 404 Not Found (аналог ResourceUnknownFault)
- Недоступность ресурса - 403 Forbidden или 401 Unauthorized в зависимости от причины (аналог ResourceUnavailableFault).

RESTful грид-сервис: преимущества

- ▶ Простота протокола.
 - Нет зависимости от сторонних средств генерации кода и сложных библиотек.
 - Легко реализовать клиент и сервис.
- ▶ Свобода выбора представлений ресурсов.
 - Возможность использования эффективных форматов данных, возможность использования различных форматов данных для одних и тех же ресурсов.
- ▶ Свобода выбора языков запросов
- ▶ Надежные механизмы восстановления ошибок связи.
 - Идемпотентные и безопасные методы: просто вызови еще раз.

RESTful грид-сервис: недостатки

- ▶ Отсутствие формальных стандартов.
 - Однако, существуют рекомендации и лучшие практики (Best Common Practice).
- ▶ Отсутствие групп сервисов.
 - Возможно, не требуются в связи с гибкой адресацией и объединением ресурсов в коллекции.
- ▶ Отсутствие уведомлений.
 - Начинают появляться общие практики, из которых могут вырасти подходящие протоколы, например PubSubHubbub.

RESTful-веб-сервисы: HTTP-методы и коллекции

- ▶ Пример адреса (URI) коллекции:
<http://example.com/resources/>
- ▶ **GET**: получение списка членов коллекции.
- ▶ **PUT**: замещение существующей коллекции новой, то есть коллекция рассматривается как самостоятельный ресурс.
- ▶ **POST**: создание нового ресурса в коллекции с автоматическим присвоением ему идентификатора.
- ▶ **DELETE**: удаление коллекции.

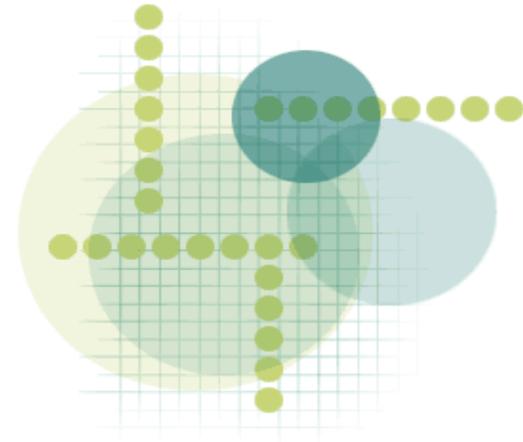
Пилот — сервис управления КОМПОЗИТНЫМИ заданиями

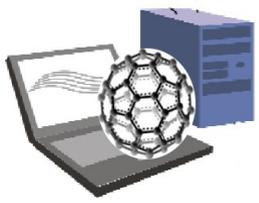
- RESTful веб-сервис
 - Входные запросы — это http(s) URI
 - GET — получить данные об объекте
 - PUT — изменить свойства объекта
 - POST — создать объект
 - DELETE — уничтожить объект
- Задания и задачи рассматриваются как JSON объекты.
- Определен развитый API и CLI



CLI Пилота

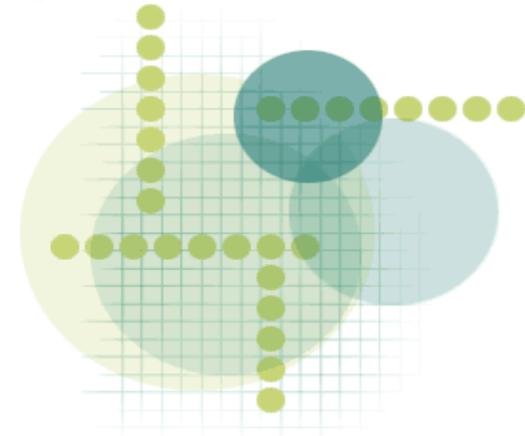
- Pilot-job-submit
- Pilot-job-cancel
- Pilot-job-status
- Pilot-task-status
- И ряд других
-

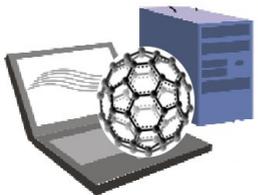




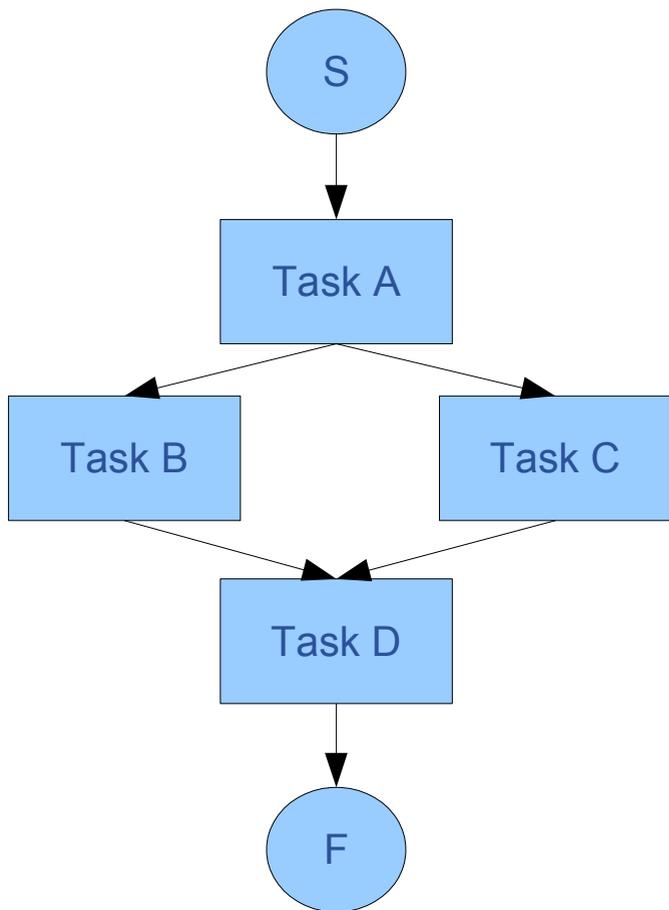
Обработка КОМПОЗИТНЫХ заданий

- Задание — совокупность задач, представленных в виде ациклического направленного графа (DAG).
- Задача запускается на исполнение если все задачи, от которых она зависит завершились успешно.

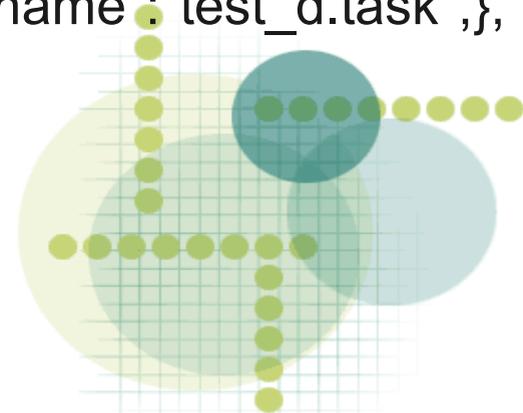




Пример композитного задания

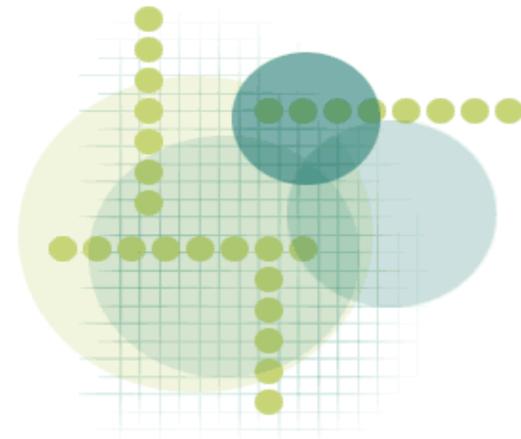


```
{ "version":2,  
  "tasks":[  
    {"id":"n0","description":"a","filename":"test_a.task",  
     "children":["n2","n1"]}  
    {"id":"n1","description":"b","filename":"test_b.task",  
     "children":["n3"]},  
    {"id":"n2","description":"c","filename":"test_c.task",  
     "children":["n3"]},  
    {"id":"n3","description":"d","filename":"test_d.task"},  
  ],  
  "requirements": [{ "hostname":  
["nanoz.grid.kiae.ru"] } ]  
}
```



Пример задачи

```
{ "version": 2,  
  "scheduler": "SLURM",  
  "queue": "ngrid",  
  "executable": "/usr/bin/whoami",  
  "stdout_file": "test.txt"  
}
```



Управление данными

- В настоящее время управление данными при обработке задания полностью возложено на пользователя.
- Для обмена данными используются gridftp серверы, доступные в ГридННС.
- Грид-шлюз использует эти серверы для загрузки файлов перед началом обработки задачи (stage-in) и выгрузки результатов (stage-out).



Модель аутентификации ГридННС.

- Модель аутентификации ГридННС построена на основе PKI с использованием цифровых сертификатов стандарта X.509.
- Выполнение заданий пользователя происходит с использованием короткоживущих прокси-сертификатов. Для выполнения задач длительностью больше чем время жизни прокси-сертификата используется механизм обновления прокси-сертификатов с помощью специального сервиса — MyProxy [26]
- Передачи прав от одного сервиса к другому производится методом делегации, что исключает передачу закрытых ключей по каналам связи.



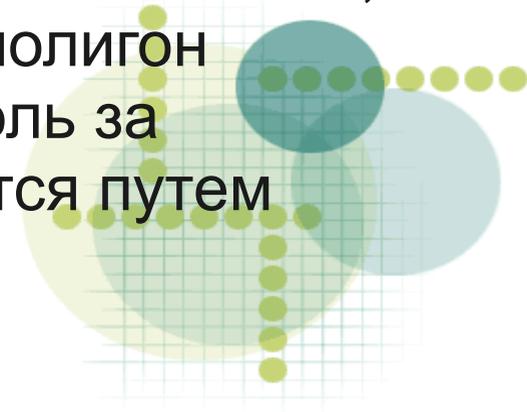
Заключение (1/2)

- Главным результатом, достигнутым в ходе реализации проекта, является обобщение RESTful веб-сервисов до полноценных грид-сервисов. В частности, был сформулирован принцип построения RESTful-грид-сервисов с использованием идемпотентных операций.
- Предложенные подходы позволили отказаться от сложных в реализации, развертывании, настройке WSRF-сервисов для ряда ключевых компонент.
- С использованием архитектурного стиля REST был реализован сервис управления композитными заданиями (workflow) и распределения задач по ресурсам — Pilot, который показал надежную работу и высокую производительность..



Заключение (2/2)

- ГридННС — это полнофункциональная грид-инфраструктура содержащая полный набор сервисов и систем, обеспечивающих ее работу. Предусмотрены службы поддержки пользователей, развернут сайт, на котором публикуются новости ГридННС, помещены многочисленные инструкции и регламенты.
- В настоящее время к ГридННС развернут на полигоне, который состоит из 11 сайтов. Суммарно полигон охватывает более пяти тысяч ЦПУ. Контроль за работоспособностью сайтов осуществляется путем регулярного запуска тестовых задач.



География ГридННС



Национальная
Нанотехнологическая
Сеть

ГридННС мониторинг

Карта

Использование

Загрузка ресурсов

Ресурсы

Ресурсы (история)

Инфраструктура

Параметры ресурсов

Сайт проекта ГридННС



*Перспективы грида
(вместо заключения)*

Перспективы

Существует множество информационных технологий ориентированных на высокопроизводительные вычисления

- Классические суперкомпьютерные вычисления
- Грид-технологии
- Облачные вычисления
- И т.д.

В чем же ниша грида? Ниша грида — это удовлетворения запроса со стороны достаточно массового пользователей на дистанционное получение вычислительных суперкомпьютерных услуг.

Перспективы

- ▶ Эффект грида тем больше, чем больше инфраструктура.
 - EGEE. Практически в любой момент пользователь может использовать очень большой вычислительный ресурс.
- ▶ Мультидисциплинарность.
 - Грид — это среда, с помощью которой пользователь получает доступ к нужным ему ресурсам. Он не привязан к конкретной области.
 - Тем не менее, наличие интерфейсов пользователя, ориентированных на конкретные области, повышает его привлекательность как платформы для вычислений.

Перспективы

- ▶ RESTful веб-сервисы
 - Простая семантика,
 - Широко применяемый протокол (http(s)).
 - Быстрая сериализация-десериализация объектов
- ▶ ГридННС
- ▶ GT-5!

Перспективы

- ▶ Виртуализация.
 - Сейчас это больше связывают с облачными вычислениями.
 - Начала применяться в грид до возникновения облачных вычислений.
- ▶ Не WSRF гриды.
 - WSRF — позволяет построить универсальный открытый грид.
 - Очень сложна в реализации
 - Как следствие имеет низкую производительность.

Спасибо за внимание!
Вопросы?

