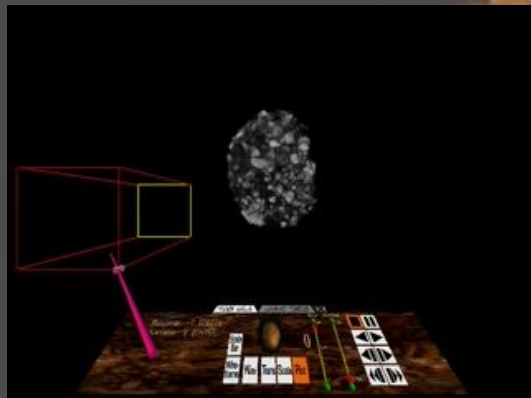
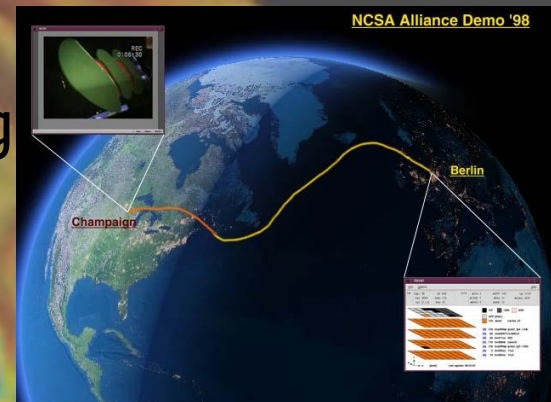


Введение в Grid Computing и Globus Toolkit™

The Globus Project™
Argonne National Laboratory
USC Information Sciences Institute



<http://www.globus.org>





Содержание

- Введение в Grid Computing
- Некоторые определения
- Архитектура Grid
- The Programming Problem
- The Globus Toolkit™
 - Введение, безопасность, управление ресурсами, управление данными
- Перспективы



Зачем оно надо? (The Grid problem)

- Flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resource

Из "The Anatomy of the Grid: Enabling Scalable Virtual Organizations"

- Дать возможность "виртуальным организациям" совместного использования географически удалённых ресурсов при совместной работе – подразумевая отсутствие...
 - центрального расположения,
 - централизованного контроля,
 - всеведения,
 - атмосферы доверия.



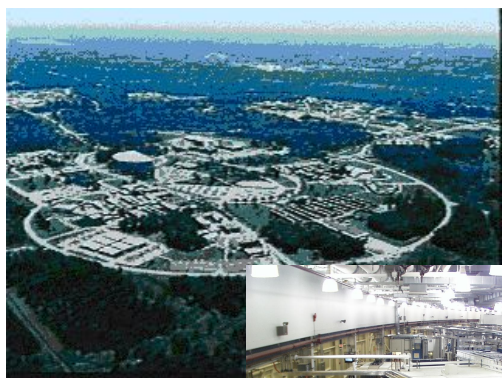
Составляющие Проблемы

- Совместное использование ресурсов
 - Компьютеры, хранение данных, сети, ...
 - Совместное использование ресурсов всегда возможно только при определённых условиях: вопросы доверия, внутренних правил, оплата, переговоры, ...
- Координированное решение задач
 - Анализ удалённых данных, вычисления, совместная работа, ...
- Виртуальные организации - динамичные, включающие различные Институты, группы
 - Научные сообщества включают различные классические организации
 - Многочисленные или нет, динамичные или статичные

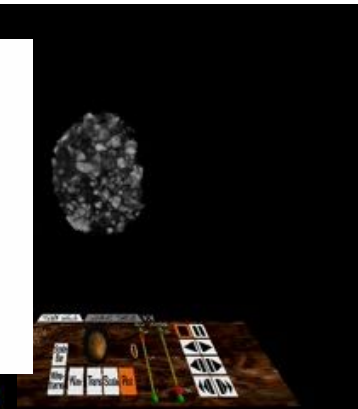


Доступ в сети к научным инструментам

Advanced Photon Source



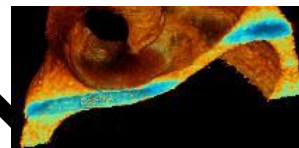
wide-area dissemination



Сбор данных в режиме реального времени



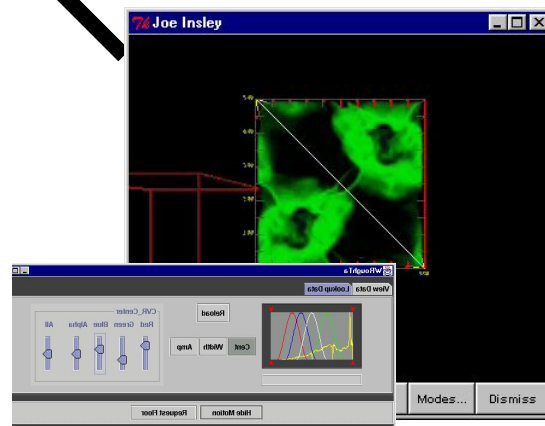
архивы



совместное управление



Томографическая реконструкция



DOE X-ray grand challenge: ANL, USC/ISI, NIST, U.Chicago

Grids в Физике Высоких энергий

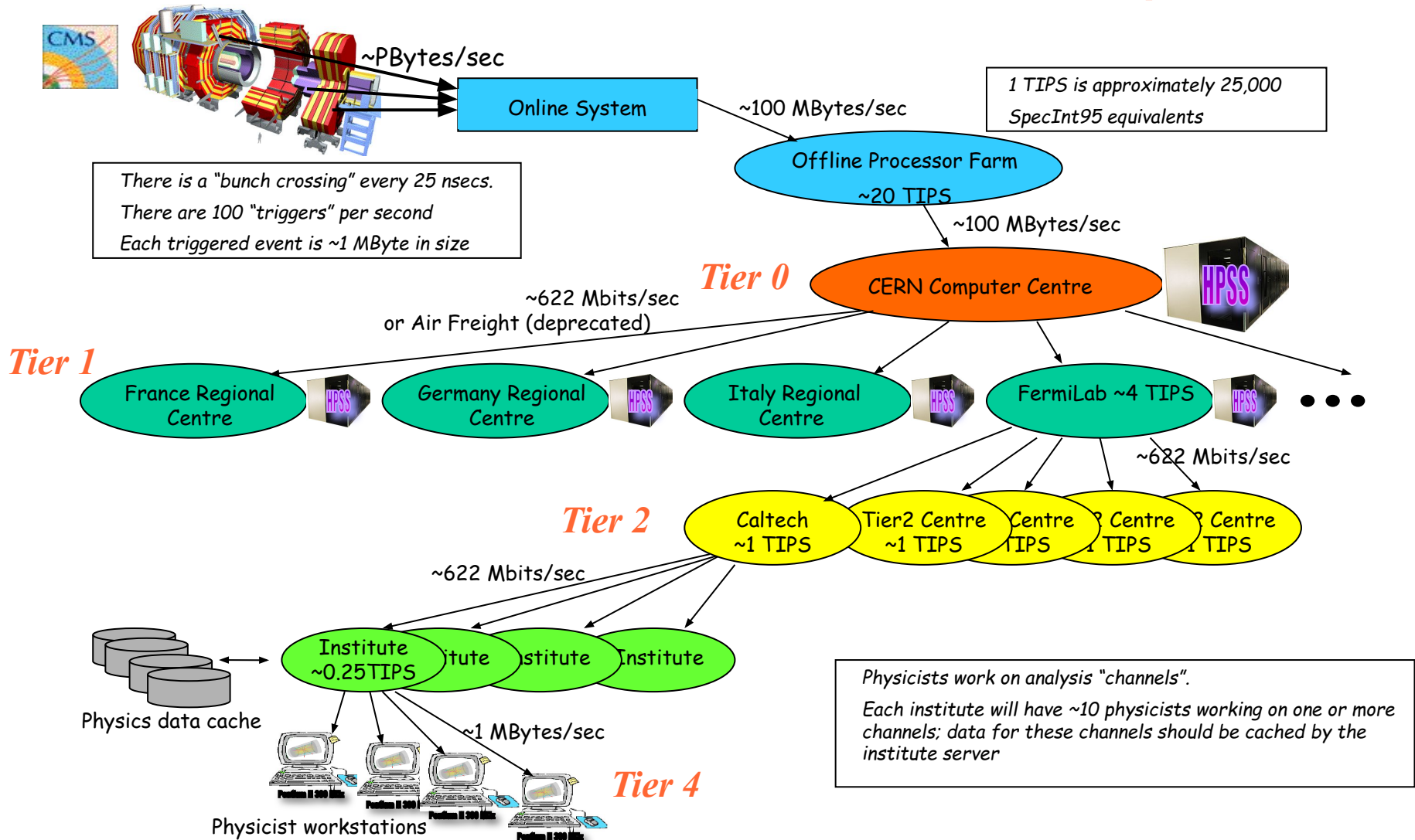


Image courtesy Harvey Newman, Caltech



Домашние компьютеры тестируют лекарства от СПИДа

Кто =

- 1000s домашних ПК
- Филантропическая компания Entropia
- Научно-исследовательская компания Scripps

Единая Цель = ускорить исследования в области СПИДа

fight AIDS @home the Olson laboratory at The Scripps Research Institute
computing toward a cure

powered by **entropia**

▶ [Fight AIDS @ Home](#)
 ▶ [The AIDS Crisis](#)
 ▶ [How Your PC can Help](#)
 ▶ [Project Status](#)
 ▶ [Get the Download](#)
 ▶ [Research Team](#)
 ▶ [The Discovery Research Team](#)
 ▶ [Links and Communities](#)
 ▶ [Entropia](#)
 ▶ [Link Your Site to FA@H](#)
 ▶ [FAQ](#)

Free Software for Your PC - By [downloading Entropia](#) onto your PC, **FightAIDS@Home** uses your computer's idle resources to accelerate powerful new anti-HIV drug design research!

FightAIDS@Home is a computational research project conducted by the [Olson laboratory](#) at [The Scripps Research Institute](#) in La Jolla, California. The project uses Entropia's global Internet computing grid, which runs both commercial and research applications on PCs.

How Your PC Helps - FightAIDS@Home uses your computer to generate and test millions of candidate drug compounds against detailed models of evolving HIV viruses, a feat previously impossible without dozens of multi-million dollar supercomputers. Every PC matters!

September 22, 2000

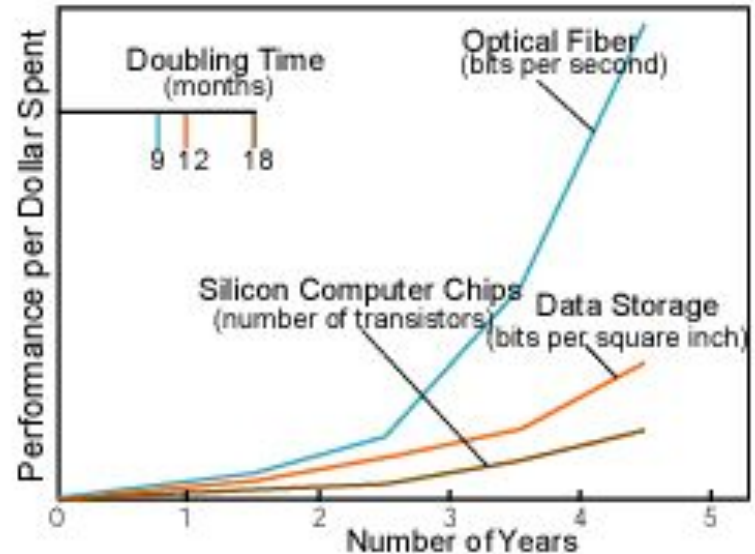
Download
Getting started is easy - [download and install](#) Entropia's free software now!

Get Project News via E-mail
Enter your email address below to receive **FightAIDS@Home** news and announcements!



Компьютерные сети

- Сети vs. Производительность компьютеров
 - Вычислительные скорости удваиваются каждые 18 месяцев
 - Скорости сетей удваиваются каждые 9 месяцев
 - Разница на целый порядок за 5 лет
- 1986 to 2000
 - компьютеры: x 500
 - сети: x 340,000
- 2001 to 2010
 - компьютеры: x 60
 - сети: x 4000



Moore's Law vs. storage improvements vs. optical improvements. Graph from **Scientific American** (Jan-2001) by Cleo Vilett, source Vined Khoslan, Kleiner, Caufield and Perkins.









The Globus Project™

Making Grid computing a reality

- Тесное сотрудничество с реальными Grid проектами в науке и промышленности
- Разработка и распространение стандартных протоколов для Grid с целью достижения совместимости и создания инфраструктуры
- Разработка и распространение стандартного программного обеспечения для Grid - универсального и мультиплатформного
- The Globus Toolkit™: Бесплатное, в прямом доступе; база для создания различных приложений и создания Grid инфраструктуры
- Global Grid Forum: Разработка стандартных протоколов и приложений для Grid



Некоторые Grid Проекты

Name	URL & Sponsors	Focus
Access Grid  <i>New</i> 	www.mcs.anl.gov/FL/accessgrid ; DOE, NSF	Create & deploy group collaboration systems using commodity technologies
BlueGrid 	IBM	Grid testbed linking IBM laboratories
DISCOM	www.cs.sandia.gov/discom DOE Defense Programs	Create operational Grid providing access to resources at three U.S. DOE weapons laboratories
DOE Science Grid  <i>New</i>	sciencegrid.org DOE Office of Science	Create operational Grid providing access to resources & applications at U.S. DOE science laboratories & partner universities
Earth System Grid (ESG)  	earthsystemgrid.org DOE Office of Science	Delivery and analysis of large climate model datasets for the climate research community
European Union (EU) DataGrid	eu-datagrid.org European Union	Create & apply an operational grid for applications in high energy physics, environmental science, bioinformatics









Некоторые Grid Проекты

Name	URL/Sponsor	Focus
EuroGrid, Grid Interoperability (GRIP) <i>New</i>	eurogrid.org European Union	Create tech for remote access to supercomp resources & simulation codes; in GRIP, integrate with Globus Toolkit™
Fusion Collaboratory <i>New</i>	fusiongrid.org DOE Off. Science	Create a national computational collaboratory for fusion research
Globus Project™	globus.org DARPA, DOE, NSF, NASA, Msoft	Research on Grid technologies; development and support of Globus Toolkit™; application and deployment
GridLab <i>New</i>	gridlab.org European Union	Grid technologies and applications
GridPP <i>New</i>	gridpp.ac.uk U.K. eScience	Create & apply an operational grid within the U.K. for particle physics research
Grid Research Integration Dev. & Support Center <i>New</i>	grids-center.org NSF	Integration, deployment, support of the NSF Middleware Infrastructure for research & education




Некоторые Grid Проекты

Name	URL/Sponsor	Focus
Grid Application Dev. Software 	hipersoft.rice.edu/grads ; NSF	Research into program development technologies for Grid applications
Grid Physics Network 	griphyn.org NSF	Technology R&D for data analysis in physics expts: ATLAS, CMS, LIGO, SDSS
Information Power Grid 	ipg.nasa.gov NASA	Create and apply a production Grid for aerosciences and other NASA missions
International Virtual Data Grid Laboratory  <i>New</i>	ivdgl.org NSF	Create international Data Grid to enable large-scale experimentation on Grid technologies & applications
Network for Earthquake Eng. Simulation Grid  <i>New</i>	neesgrid.org NSF	Create and apply a production Grid for earthquake engineering
Particle Physics Data Grid 	ppdg.net DOE Science	Create and apply production Grids for data analysis in high energy and nuclear physics experiments



Некоторые Grid Проекты

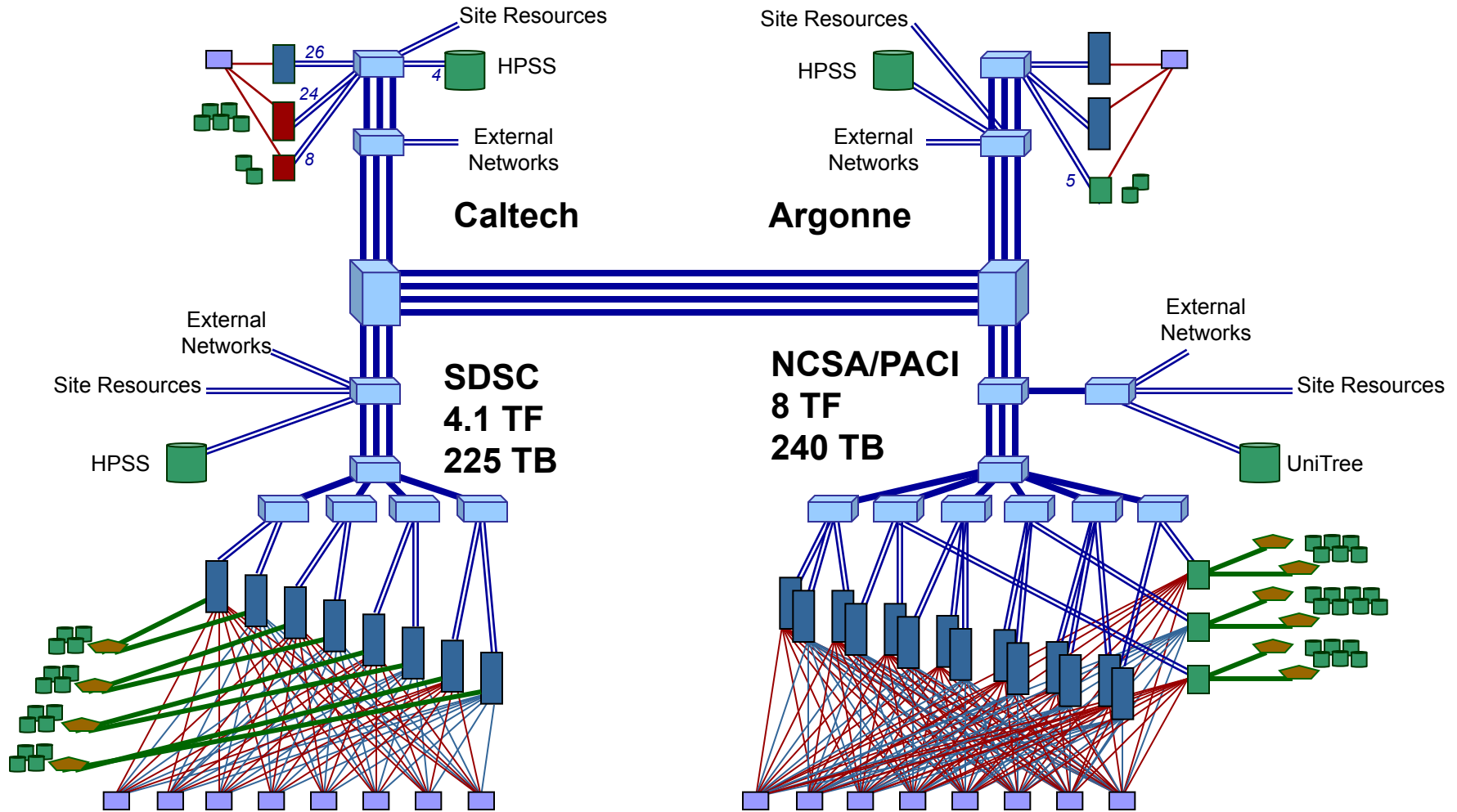
Name	URL/Sponsor	Focus
TeraGrid  <i>New</i>	teragrid.org NSF	U.S. science infrastructure linking four major resource sites at 40 Gb/s
UK Grid Support Center  <i>New</i>	grid-support.ac.uk U.K. eScience	Support center for Grid projects within the U.K.
Unicore	BMBFT	Technologies for remote access to supercomputers

Also many technology R&D projects:
e.g., Condor, NetSolve, Ninf, NWS

See also www.gridforum.org

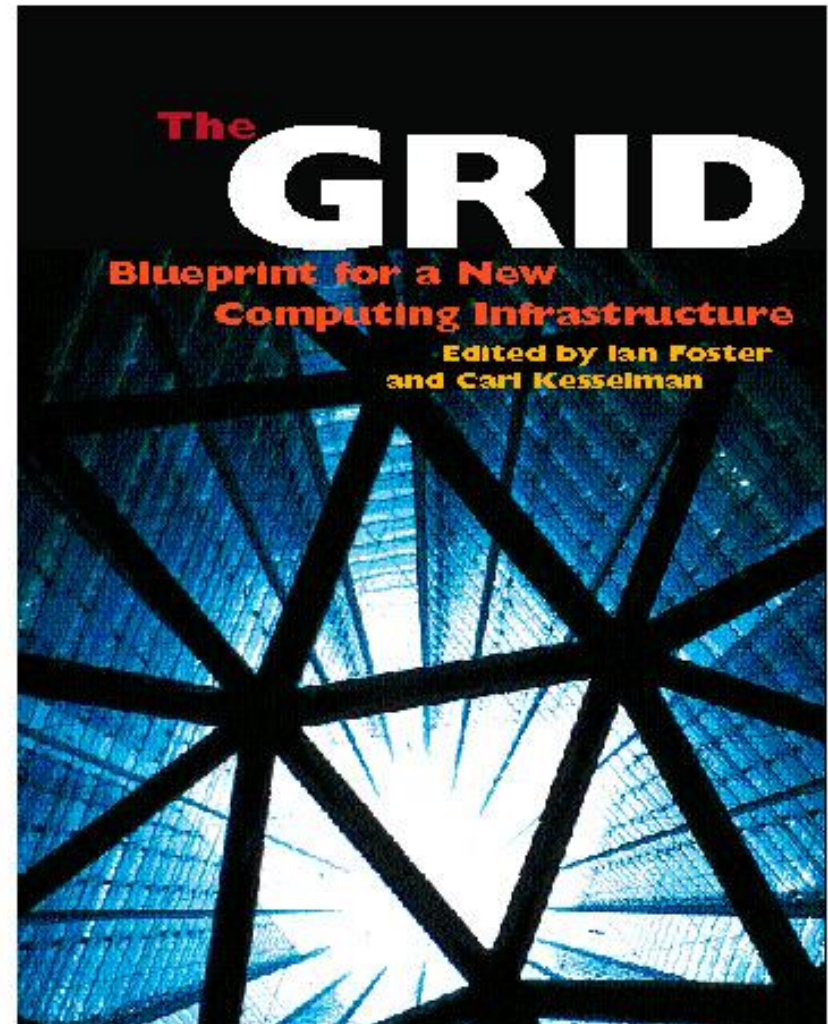


The 13.6 TF TeraGrid: Computing at 40 Gb/s



Для информации

- Globus Project™
 - www.globus.org
- Grid Forum
 - www.gridforum.org
- Книга (Morgan Kaufman)
 - www.mkp.com/grids





Некоторые определения

The Globus Project™

Argonne National Laboratory
USC Information Sciences Institute

<http://www.globus.org>



Некоторые важные определения

- Ресурс
- Протокол сети
- Сервис, обеспечиваемый сетью
- Интерфейс приложения - Application Programmer Interface (API)
- Software Development Kit (SDK)
- Синтаксис



Ресурс

- Всё, что можно использовать совместно
 - Компьютеры, накопители информации, данные, компьютерные программы и т.д.
- Не обязательно должен быть физической единицей
 - Condor pool, distributed file system, ...
- Определяется интерфейсами, а не устройствами
 - 'планировщик' (such as LSF and PBS) определяет компьютерный ресурс
 - Open/close/read/write определяет доступ к распределённой системе файлов , e.g. NFS, AFS, DFS

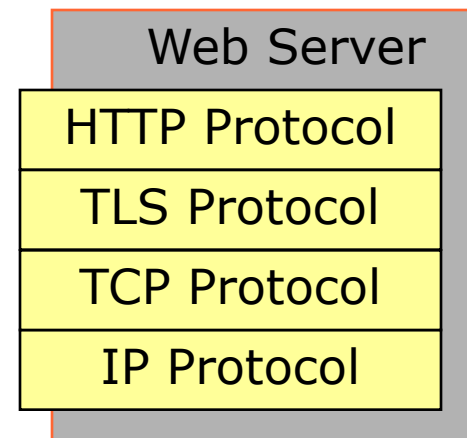
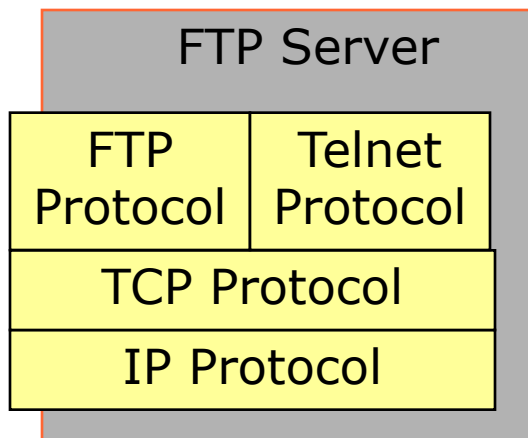


Сетевой протокол

- Формальное описание форматов сообщений и набор правил для обмена сообщениями
 - Правила могут определять последовательность обмена сообщениями
 - Протокол может определять изменение состояния системы в конечной точке (например, изменение состояния системы файлов)
- Хорошие протоколы созданы с одной целью
 - Протоколы можно накладывать друг на друга
- Примеры Протоколов
 - IP, TCP, TLS (было SSL), HTTP, Kerberos

Сервис, обеспечиваемый сетью

- Создание протокола, который определяет набор возможностей
 - Протокол определяет связь с сервисом
 - Все сервисы нуждаются в протоколе
 - Не все протоколы используются для предоставления сервиса(e.g. IP, TLS)
- Примеры: FTP и Web серверы





Application Programming Interface

- Набор спецификаций для приложения
 - Относится к функциональному определению, а не к конкретному воплощению
 - Например, существует много воплощений MPI
- Часто эти спецификации бывают привязаны к конкретному языку программирования
 - Название программы, количество и тип аргументов, определённые языковые конструкции
 - Поведение функции или программы
- Примеры
 - GSS API (security), MPI (message passing)



Протокол может иметь множество APIs

- TCP/IP APIs включают в себя BSD sockets, Winsock, System V streams, ...
- Протокол предоставляет совместимость: программы, использующие разные APIs, могут обмениваться информацией
- Мне не нужно знать API другого



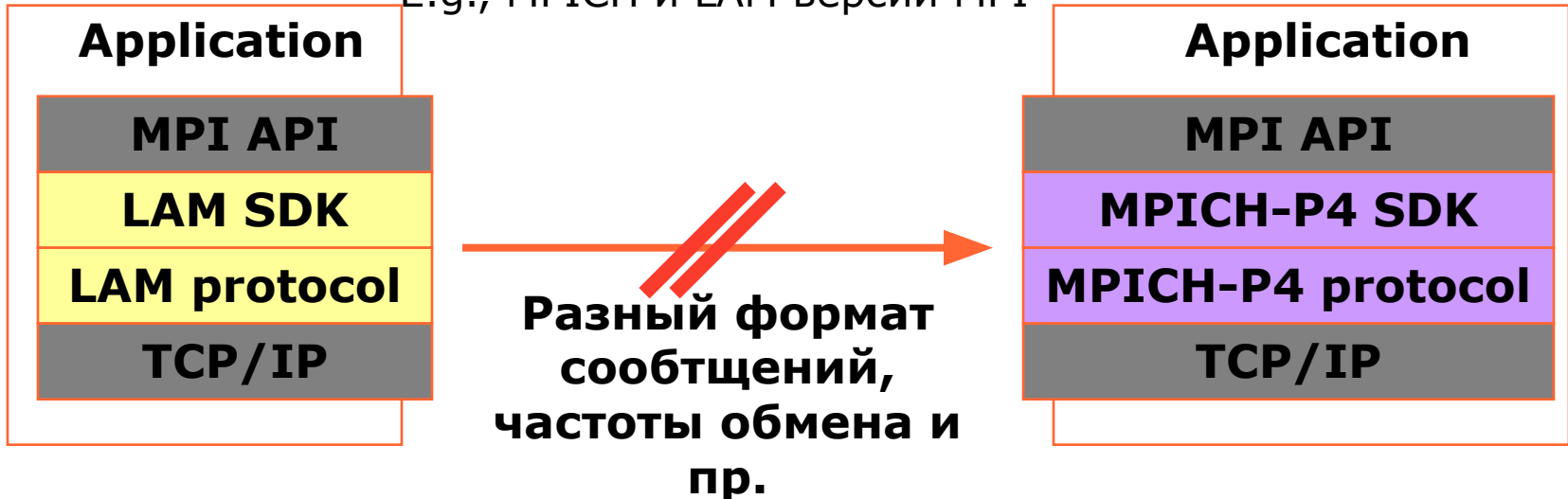


API может иметь много протоколов

- MPI - портативно: любая правильная программа должна компилироваться и работать на любой платформе
- Но никакой совместимости на уровне SDK

-

E.g., MPICH и LAM версии MPI





APIs и Протоколы очень важны

- Стандартные APIs/SDKs важны
 - Они дают приложению *портативность*
 - Но без стандартных протоколов внутренняя совместимость невозможна (любой SDK понимает любой протокол?)
- Стандартные протоколы важны
 - Дают внутреннюю совместимость независимости от месторасположения
 - Делают возможным совместные инфраструктуры
 - Но без стандартных APIs/SDKs становится невозможным портативность приложения (различные платформы работают с протоколами по-разному)



Software Development Kit

- Определённое воплощение API
- SDK состоит из библиотек и программ
 - Представляет собой воплощение спецификаций API
- Для одного API может быть много SDKs
- Примеры SDK
 - MPICH, Motif Widgets, LAM



Синтаксис

- Правила для расшифровки информации
 - XML, Condor ClassAds, Globus RSL
 - X.509 certificate format (RFC 2459)
 - Cryptographic Message Syntax (RFC 2630)
- Просьба, не путать с протоколом!
 - Один и тот же синтаксис может быть использован разными протоколами (e.g., XML)
- Синтаксис может быть наложен один на другой
 - E.g., Condor ClassAds -> XML -> ASCII
 - Очень важно понимать концепцию наложения синтаксиса при сравнениях и оценке.



Архитектура Grid

The Globus Project™

Argonne National Laboratory
USC Information Sciences Institute

<http://www.globus.org>



Зачем обсуждать Архитектуру?

- Описание
 - Предложить общие термины для обсуждения Grid систем
- Направление работ
 - Определить основные области, требующие создания сервиса
- Предопределение
 - Определить стандартные "Intergrid"овские протоколы и APIs для создания совместимых и портативных приложений



Некоторые Требования

- Идентификация
- Авторизация&правила
- Поиск ресурсов
- Описание ресурсов
- Резервирование ресурсов
- Распределённые алгоритмы
- Доступ к удалённым данным
- Высоко-скоростная пересылка данных
- Гарантирование производительности
- Обнаружение несанкционированного доступа
- Распределение ресурсов
- Счета и оплата
- Обнаружение неполадок
- Эволюция систем
- Мониторинг
- И т.д.
- И т.д.
- ...



Три пункта для превращения "Grid computing" в рутину...

1) Новые подходы к решению проблем

- Data Grids, распределённые вычисления, peer-to-peer, объединённые grids, ...

2) Структурирование и написание программ

- Абстракции, инструменты

3) Обеспечение совместного доступа к ресурсам

- Инспекция ресурсов, доступ, бронирование, выделение; аутентификация, авторизация; коммуникация; Диагностика сбоев; ...

Programming Problem

Systems Problem



В итоге, Grid Архитектура, ориентированная на Протоколы:

- Создание протоколов и сервисной оболочки Grid
 - Доступ к удалённым ресурсам через протоколы
 - Новые сервисы: предоставление ресурсов
 - “работать в Grid” = понимать Intergrid протоколы
 - В основном уже имеющиеся протоколы или их расширения
- Создание Grid APIs & SDKs
 - Интерфейсы к Grid протоколам и сервисной оболочке
 - Помощь в создании приложений путём создания абстракций на более высоком уровне
- Модель , имеющая огромный успех - Internet

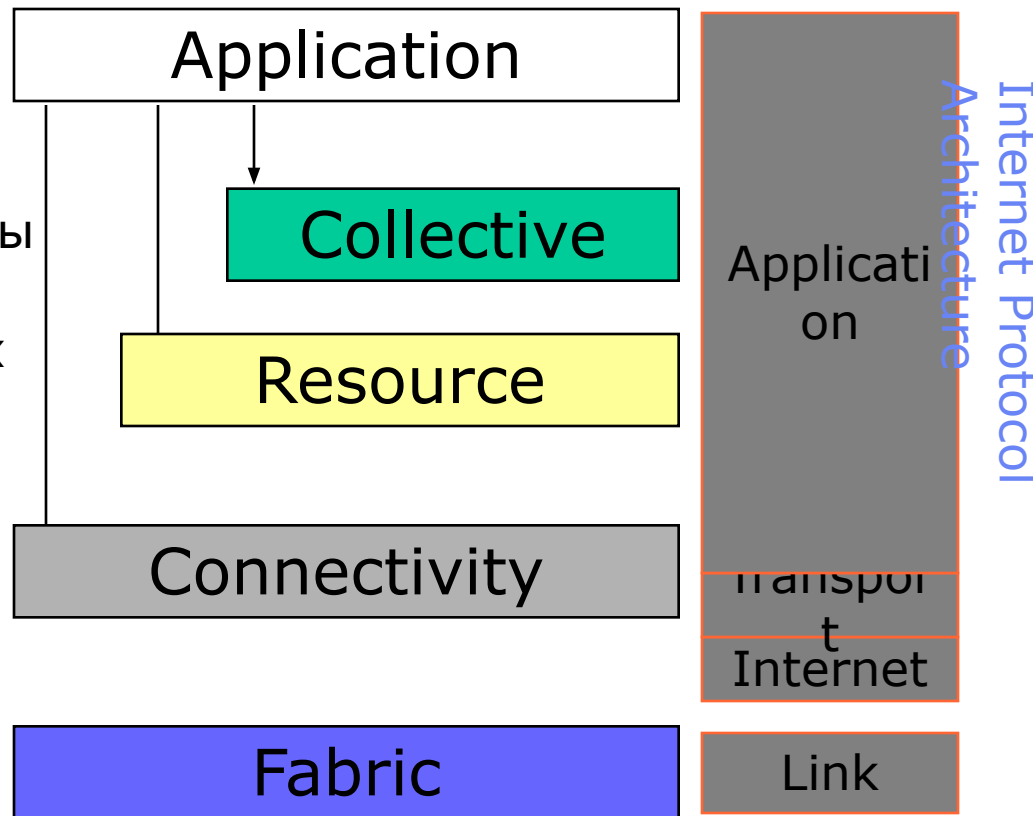
Многоуровневая Архитектура Grid (По Аналогии с Архитектурой Интернета)

“Координация многочисленных ресурсов”: специфические сервисы

“Совместное использование одних ресурсов”: доступ по договору, использование под контролем

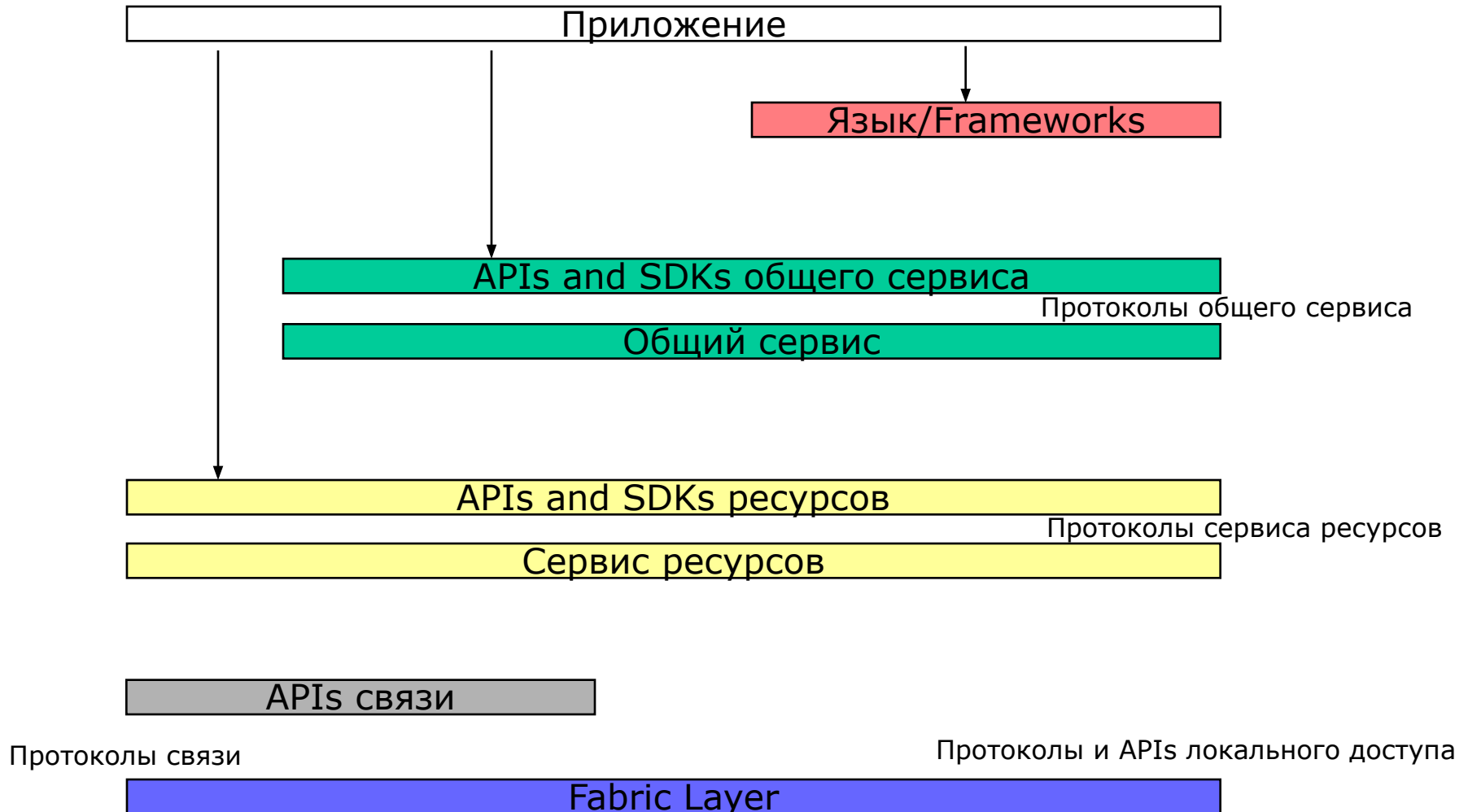
“Коммуникация”: коммуникация (Internet протоколы) & защищённость

“локальный контроль над ресурсами”: Доступ и контроль ресурсов





Протоколы, Сервис и APIs находятся на каждом уровне





Важные моменты:

- Основано на протоколах и сервисе Интернет
 - Связь, маршруты, определение имени, и т.д.
- “Многоуровневость” здесь чисто концептуальна, НЕ накладывает никаких ограничений на то, кто какие функции может вызвать
 - Протоколы/сервис/APIs/SDKs в идеале, будут самодостаточны
 - Некоторые вещи здесь фундаментальны: например, коммуникация и защищённость
 - Привлекательно для функций высокого уровня использовать стандартные функции низкого уровня



Ну и что сейчас с Архитектурой?

- Не существует никаких 'официальных' стандартов
- НО:
 - Globus Toolkit™ является практически de facto стандартом для многих важных протоколов (связь, ресурсы и общие)
 - GGF имеет рабочую группу по архитектуре
 - Технические детали находятся сейчас в разработке: защищённость, управление ресурсами и данными, информационный сервис
 - Документы (в области безопасности) приняты к публикации в Интернете



'Fabric' уровень Протоколы и сервис

- Всё что можно ожидать: огромное разнообразие совместных ресурсов
 - ПК, файловые системы, архивы, каталоги метаданных, сети, сенсоры и т.д, и т.п.
- Несколько ограничений на технологии низких уровней (Few constraints on low-level technology): протоколы связи и ресурсов являются узким местом
- Определяется интерфейсами, а не физическими характеристиками



Уровень связи: Протоколы & Сервис

- Коммуникация
 - Internet протоколы: IP, DNS, routing, etc.
- Защищённость: Grid Security Infrastructure (GSI)
 - Единая идентификация, авторизация и защищённая передача сообщений
 - Однократный логин, делегирование, идентификация
 - Public key technology, SSL, X.509, GSS-API
 - Инфраструктура поддержки: централизованная выдача сертификатов, управление сертификатами и ключами, ...

GSI: www.gridforum.org/security/gsi



Уровень ресурсов: Протоколы & Сервис

- Grid Resource Allocation Management (GRAM)
 - Удалённые ресурсы : выделение, резервирование, мониторинг и управление компьютерными ресурсами
 - GridFTP протокол (FTP расширения)
 - Высокоскоростной доступ к данным и пересылка
- Grid Resource Information Service (GRIS)
 - Доступ к информации
- В проекте: доступ к каталогам, доступ к библиотеке програм, Catalog access, code repository access, и т. Д.
- Всё построено на уровне: GSI & IP

GRAM, GridFTP, GRIS: www.globus.org



Общий Уровень: Протоколы & Сервис

- Распределение ресурсов (e.g., Condor Matchmaker)
 - Поиск и выявление ресурсов
- Каталог реплик
- Сервис копирования
- Сервис по одновременному резервированию и выделению
- И т.д.



Пример: Data Grid Архитектура

App

Приложение, специфичное для какой-то области

Collective (App)

Выбор реплики, управление заданием, виртуальный каталог данных, ...

Collective (Generic)

Каталог реплик, управление репликами, выделение ресурсов, выдача сертификатов, каталоги метаданных

Resource

Доступ к данным, доступ к компьютерам, доступ к информации о сети, ..

Connect

Коммуникации, поиск сервиса (DNS), идентификация, авторизация, делегация

Fabric

Системы хранения данных, кластеры, сети, ...



The Programming Problem

The Globus Project™

Argonne National Laboratory
USC Information Sciences Institute

<http://www.globus.org>



The Programming Problem

- But how do I develop robust, secure, long-lived, well-performing applications for dynamic, heterogeneous Grids?
- I need, presumably:
 - Abstractions and models to add to speed/robustness/etc. of development
 - Tools to ease application development and diagnose common problems
 - Code/tool sharing to allow reuse of code components developed by others



Проблема программирования

- Ну и как мне создать надёжное, долговременное, высокоэффективное приложение для динамичных и разнородных Grids?
- Для этого мне нужно:
 - Абстракции и модели чтобы ускорить/улучшить сам процесс
 - Набор программных средств для диагностики проблем и упрощения написания программы
 - Создать универсальные средства, чтобы было возможно использование некоторых компонент другими



Examples of Grid Programming Technologies

- MPICH-G2: Grid-enabled message passing
- CoG Kits, GridPort: Portal construction, based on N-tier architectures
- GDMP, Data Grid Tools, SRB: replica management, collection management
- Condor-G: workflow management
- Legion: object models for Grid computing
- Cactus: Grid-aware numerical solver framework
 - Note tremendous variety, application focus

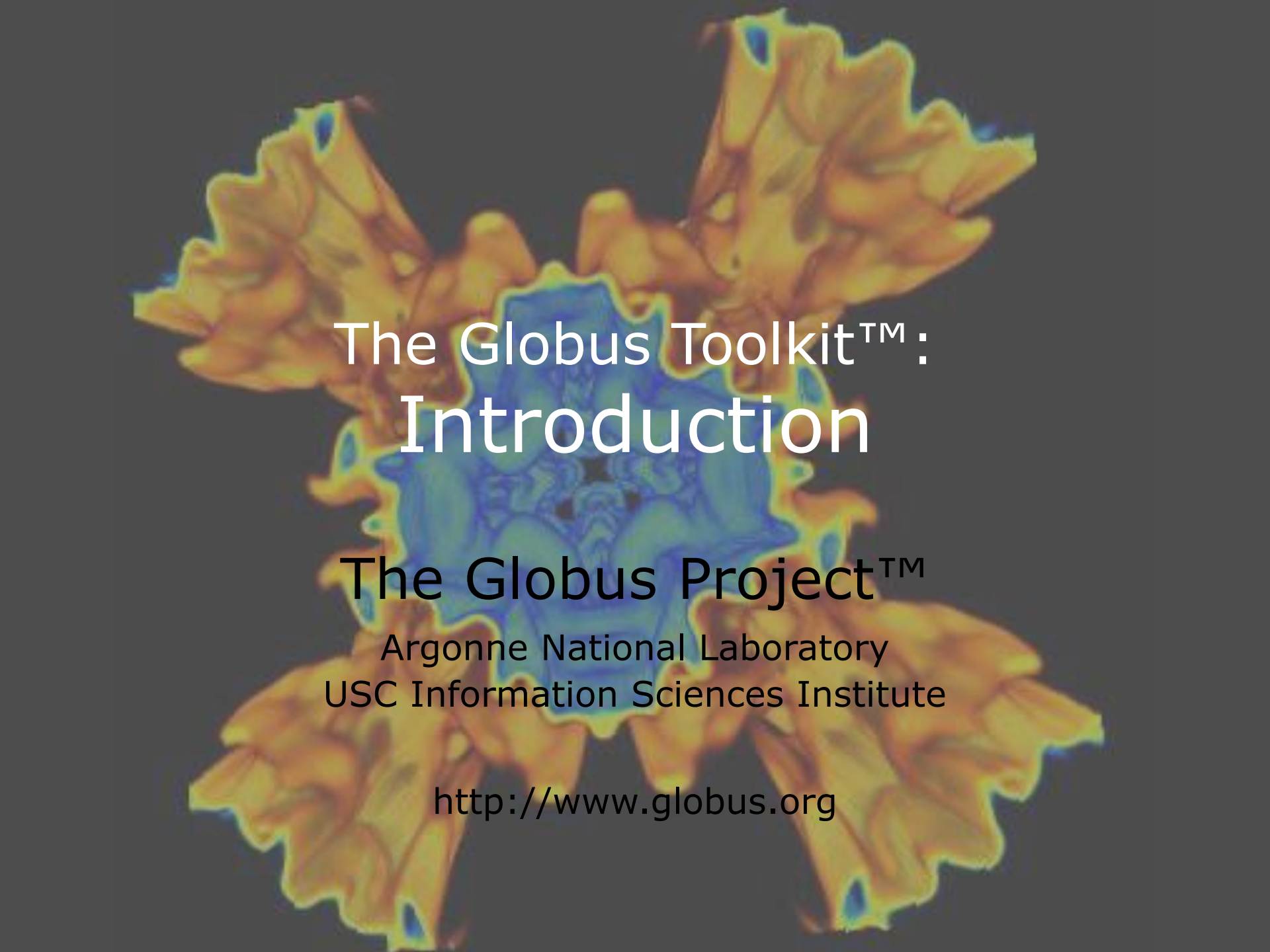


Примеры Программных Технологий в Grid

- MPICH-G2: адаптированный для Grid MPI
- CoG Kits, GridPort: идея портала, основано на N-уровневой архитектуре
- GDMP, Data Grid Tools, SRB: управление репликами, набором данных
- Condor-G: управление процессом расчётов
- Legion: объектные модели для программирования в Grid
- Cactus: адаптированные для Grid набор средств для решения численных задач
 - Следует обратить внимание на огромное разнообразие средств (все ориентированны на определённое приложение)

За всем этим стоит единый набор программных средств

- Ни один из перечисленных проектов не создавал протоколы и пр. с нуля!
- Использовался единый набор средств, который...
 - Имеет все основные функции
 - > SDKs который может быть использован для создания различных программных продуктов
 - > Стандартный сервис, который легко установить
 - Надёжный, правильно спроектированный, не противоречащий себе
 - Является бесплатным, широко доступным
- Всем этим требованиям отвечает Globus Toolkit™...



The Globus Toolkit™: Introduction

The Globus Project™

Argonne National Laboratory
USC Information Sciences Institute

<http://www.globus.org>



Globus Toolkit™

- Набор программных средств, решающий основные технические проблемы при создании программного обеспечения для Grid
 - Предлагает 'пакетный' набор средств
 - Позволяет *поэтапное* создание программных средств и приложений для Grid
 - Воплощает стандартные Grid протоколы и APIs
 - Доступен бесплатно для всех (Open source)



Общий подход

- Определить Grid протоколы & APIs
 - Доступ к удалённым ресурсам посредством протоколов
 - Интегрировать и расширить имеющиеся стандарты
- Создать соответствующий набор средств
 - Доступный всем Globus Toolkit
 - Набор утилит, SDKs, сервис, и т.д.
- Адаптировать для Grid множество известных приложений
 - Globus Toolkit, FTP, SSH, Condor, SRB, MPI, ...
- Учиться на своём опыте

(Эти тезисы взяты с официального сайта...)



Четыре ключевых протокола

- The Globus Toolkit™ основан на четырёх основных протоколах
 - Уровень связи:
 - > *защищённость*: Grid Security Infrastructure (GSI)
 - Уровень ресурсов:
 - > *Управление ресурсами*: Grid Resource Allocation Management (GRAM)
 - > *Информационный сервис*: Grid Resource Information Protocol (GRIP)
 - > *Пересылка данных*: Grid File Transfer Protocol (GridFTP)
- Также основные протоколы 'общего' уровня
 - Информационный сервис, управление репликами, и т.д.



The Globus Toolkit™: Security Services

The Globus Project™

Argonne National Laboratory
USC Information Sciences Institute

<http://www.globus.org>

Проблемы при реализации Grid Security...

- Resources being used may be valuable & the problems being solved sensitive
- Resources are often located in distinct administrative domains
 - Each resource has own policies & procedures
- Set of resources used by a single computation may be large, dynamic, and unpredictable
 - Not just client/server, requires delegation
- It must be broadly available & applicable
 - Standard, well-tested, well-understood protocols; integrated with wide variety of tools



GSI in Action

"Create Processes at A and B that Communicate & Access Files at C"

User

Single sign-on via "grid-id"
& generation of proxy cred.

Or: retrieval of proxy cred.
from online repository

User Proxy

Proxy
credential

Remote process
creation requests*

Site A
(Kerberos)

GSI-enabled
GRAM server

Authorize
Map to local id
Create process
Generate credentials

Ditto

GSI-enabled
GRAM server

Site B
(Unix)

Computer

Process

Local id

Kerberos
ticket

Restricted
proxy

Communication*

Remote file
access request*

Computer

Process

Local id

Restricted
proxy

Site C
(Kerberos)

GSI-enabled
FTP server

Authorize
Map to local id
Access file

Storage
system

* With mutual authentication



Grid Security Requirements

User View

- 1) Easy to use
- 2) Single sign-on
- 3) Run applications
ftp,ssh,MPI,Condor,Web,...
- 4) User based trust model
- 5) Proxies/agents
(delegation)

Resource Owner View

- 1) Specify local access control
- 2) Auditing, accounting, etc.
- 3) Integration w/ local system
Kerberos, AFS, license mgr.
- 4) Protection from
compromised
resources

Developer View

API/SDK with authentication, flexible message protection,
flexible communication, delegation, ...

Direct calls to various security functions (e.g. GSS-API)

Or security integrated into higher-level SDKs:

E.g. GlobusIO, Condor-G, MPICH-G2, HDF5, etc.



X.509 Proxy Certificate

- Defines how a short term, restricted credential can be created from a normal, long-term X.509 credential
 - A “proxy certificate” is a special type of X.509 certificate that is signed by the normal end entity cert, or by another proxy
 - Supports single sign-on & delegation through “impersonation”
 - Currently an IETF draft



Globus Security APIs

- Generic Security Service (GSS) API
 - IETF standard
 - Provides functions for authentication, delegation, message protection
 - Decoupled from any particular communication method
- But GSS-API is somewhat complicated, so we also provide the easier-to-use `globus_gss_assist` API.
- GSI-enabled SASL is also provided



Results

- GSI adopted by 100s of sites, 1000s of users
 - Globus CA has issued >3000 certs (user & host), >1500 currently active; other CAs active
- Rollouts are currently underway all over:
 - NSF Teragrid, NASA Information Power Grid, DOE Science Grid, European Data Grid, etc.
- Integrated in research & commercial apps
 - GrADS testbed, Earth Systems Grid, European Data Grid, GriPhyN, NEESgrid, etc.
- Standardization begun in Global Grid Forum, IETF



GSI Applications

- Globus Toolkit™ uses GSI for authentication
- Many Grid tools, directly or indirectly, e.g.
 - Condor-G, SRB, MPICH-G2, Cactus, GDMP, ...
- Commercial and open source tools, e.g.
 - ssh, ftp, cvs, OpenLDAP, OpenAFS
 - SecureCRT (Win32 ssh client)
- And since we use standard X.509 certificates, they can also be used for
 - Web access, LDAP server access, etc.



Ongoing and Future GSI Work

- Protection against compromised resources
 - Restricted delegation, smartcards
- Standardization
- Scalability in numbers of users & resources
 - Credential management
 - Online credential repositories (“MyProxy”)
 - Account management
- Authorization
 - Policy languages
 - Community authorization



Security Summary

- GSI successfully addresses wide variety of Grid security issues
- Broad acceptance, deployment, integration with tools
- Standardization on-going in IETF & GGF
- Ongoing R&D to address next set of issues
- For more information:
 - www.globus.org/research/papers.html
 - > “A Security Architecture for Computational Grids”
 - > “Design and Deployment of a National-Scale Authentication Infrastructure”
 - www.gridforum.org/security



The Globus Toolkit™:
Resource Management
Services

The Globus Project™

Argonne National Laboratory
USC Information Sciences Institute

<http://www.globus.org>



The Challenge

- Enabling secure, controlled remote access to heterogeneous computational resources and management of remote computation
 - Authentication and authorization
 - Resource discovery & characterization
 - Reservation and allocation
 - Computation monitoring and control
- Addressed by new protocols & services
 - GRAM protocol as a basic building block
 - Resource brokering & co-allocation services
 - GSI for security, MDS for discovery



Resource Management

- The Grid Resource Allocation Management (GRAM) protocol and client API allows programs to be started on remote resources, despite local heterogeneity
- Resource Specification Language (RSL) is used to communicate requirements
- A layered architecture allows application-specific resource brokers and co-allocators to be defined in terms of GRAM services
 - Integrated with Condor, PBS, MPICH-G2, ...

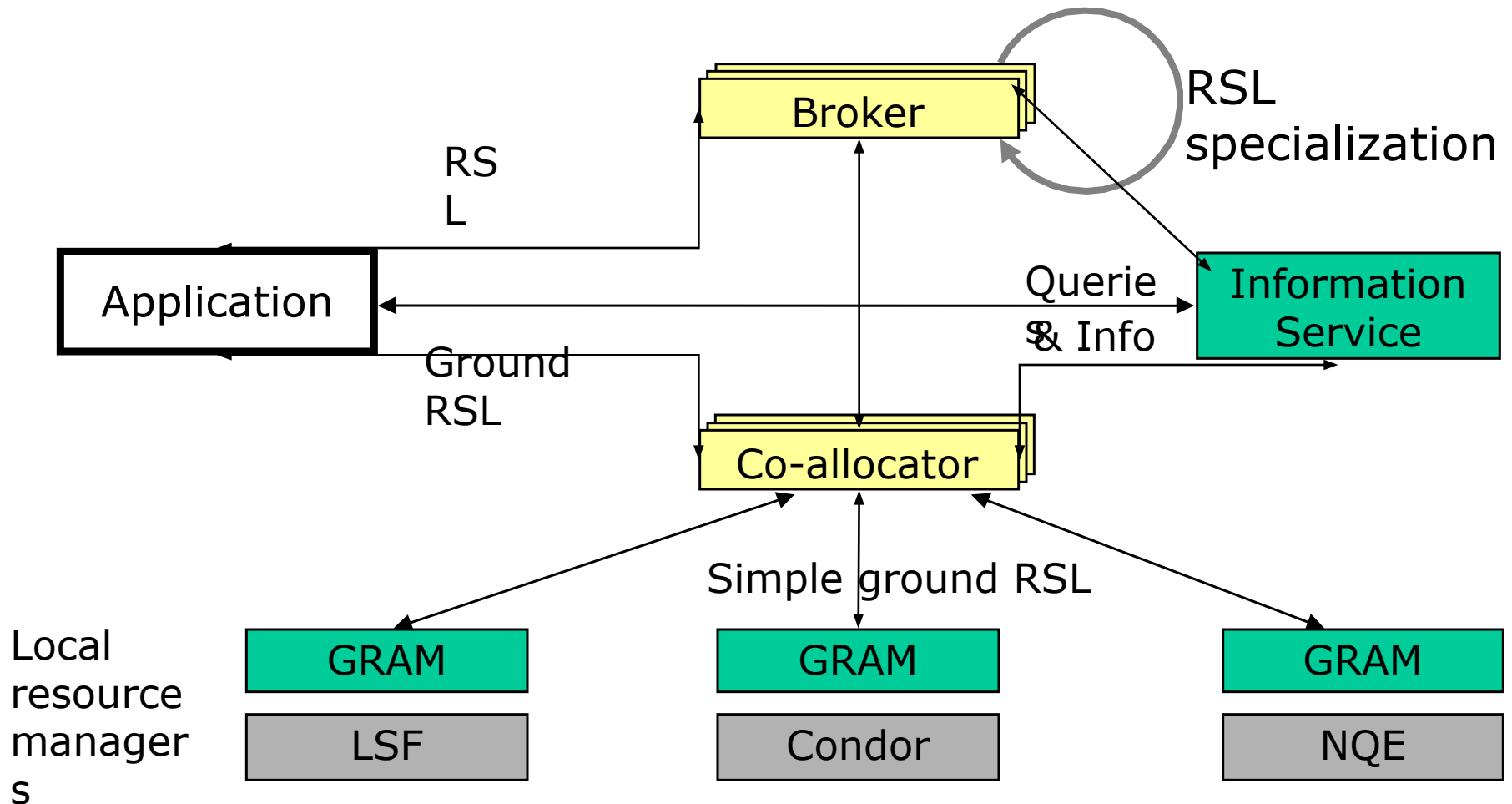


GRAM Protocol

- GRAM-1: Simple HTTP-based RPC
 - Job request
 - > Returns a “job contact”: Opaque string that can be passed between clients, for access to job
 - Job cancel, status, signal
 - Event notification (callbacks) for state changes
 - > Pending, active, done, failed, suspended
- GRAM-1.5 (U Wisconsin contribution)
 - Add reliability improvements
 - > Once-and-only-once submission
 - > Recoverable job manager service
 - > Reliable termination detection
- GRAM-2: Moving to Web Services (SOAP)...



Resource Management Architecture





Resource Specification Language

- Common notation for exchange of information between components
 - Syntax similar to MDS/LDAP filters
- RSL provides two types of information:
 - Resource requirements: Machine type, number of nodes, memory, etc.
 - Job configuration: Directory, executable, args, environment
- Globus Toolkit provides an API/SDK for manipulating RSL



RSL Syntax

- Elementary form: parenthesis clauses
 - (attribute op value [value ...])
- Operators Supported:
 - <, <=, =, >=, >, !=
- Some supported attributes:
 - executable, arguments, environment, stdin, stdout, stderr, resourceManagerContact, resourceManagerName
- Unknown attributes are passed through
 - May be handled by subsequent tools



Constraints: "&"

- For example:
& (count>=5) (count<=10)
 (max_time=240) (memory>=64)
 (executable=myprog)
- "Create 5-10 instances of **myprog**, each on a machine with at least 64 MB memory that is available to me for 4 hours"



Disjunction: “|”

- For example:
 & (executable=myprog)
 (| (&(count=5)(memory>=64))
 (&(count=10)(memory>=32)))
- Create 5 instances of myprog on a machine that has at least 64MB of memory, or 10 instances on a machine with at least 32MB of memory

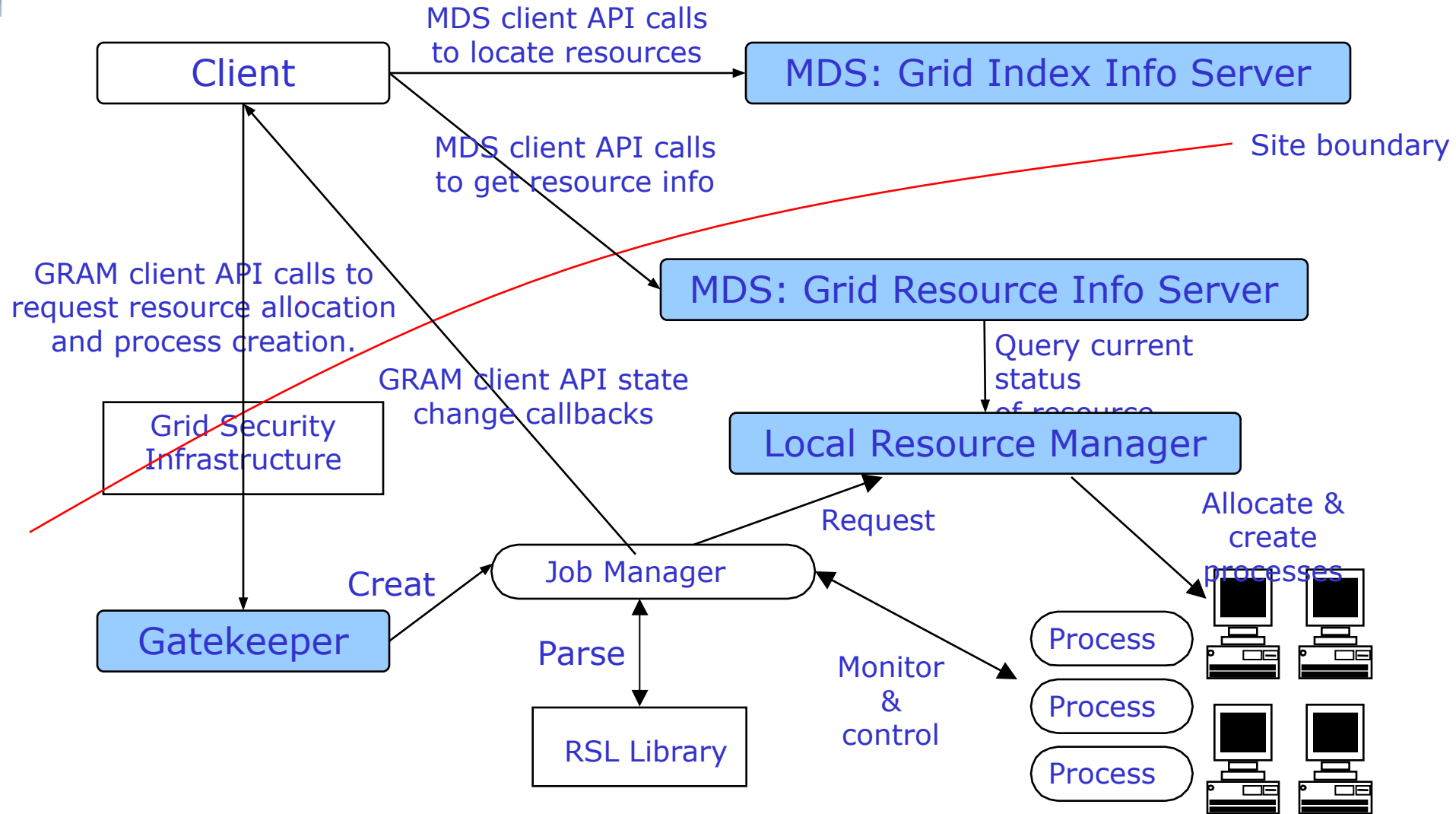


Globus Toolkit Implementation

- Gatekeeper
 - Single point of entry
 - Authenticates user, maps to local security environment, runs service
 - In essence, a “secure inetd”
- Job manager
 - A gatekeeper service
 - Layers on top of local resource management system (e.g., PBS, LSF, etc.)
 - Handles remote interaction with the job



GRAM Components





Co-allocation

- Simultaneous allocation of a resource set
 - Handled via optimistic co-allocation based on free nodes or queue prediction
 - In the future, advance reservations will also be supported (already in prototype)
- Globus APIs/SDKs support the co-allocation of specific multi-requests
 - Uses a Globus component called the Dynamically Updated Request Online Co-allocator (DUROC)



Multirequest: "+"

- A multirequest allows us to specify multiple resource needs, for example
 - + (& (count=5)(memory>=64)
(executable=p1))
(&(network=atm) (executable=p2))
 - Execute 5 instances of p1 on a machine with at least 64M of memory
 - Execute p2 on a machine with an ATM connection
- Multirequests are central to co-allocation



A Co-allocation Multirequest

```
+( & (resourceManagerContact=  
  "flash.isi.edu:754:/C=US/.../CN=flash.isi.edu-fork")  
  (count=1)  
  (label="subjob A")  
  (executable=my_app1)  
)  
( & (resourceManagerContact=  
  "sp139.sdsc.edu:8711:/C=US/.../CN=sp097.sdsc.edu-lsf")  
  (count=2)  
  (label="subjob B")  
  (executable=my_app2)  
)
```

Different resource managers

Different counts

Different executables



The Globus Toolkit™: Information Services

The Globus Project™

Argonne National Laboratory
USC Information Sciences Institute

<http://www.globus.org>



Grid Information Services

- System information is critical to operation of the grid and construction of applications
 - What resources are available?
 - > Resource discovery
 - What is the “state” of the grid?
 - > Resource selection
 - How to optimize resource use
 - > Application configuration and adaptation?
- We need a general information infrastructure to answer these questions



Examples of Useful Information

- Characteristics of a compute resource
 - IP address, software available, system administrator, networks connected to, OS version, load
- Characteristics of a network
- Bandwidth and latency, protocols, logical topology
- Characteristics of the Globus infrastructure
 - Hosts, resource managers



Grid Information: Facts of Life

- Information is always old
 - Time of flight, changing system state
 - Need to provide quality metrics
- Distributed state hard to obtain
 - Complexity of global snapshot
- Component will fail
- Scalability and overhead
- Many different usage scenarios
 - Heterogeneous policy, different information organizations, etc.

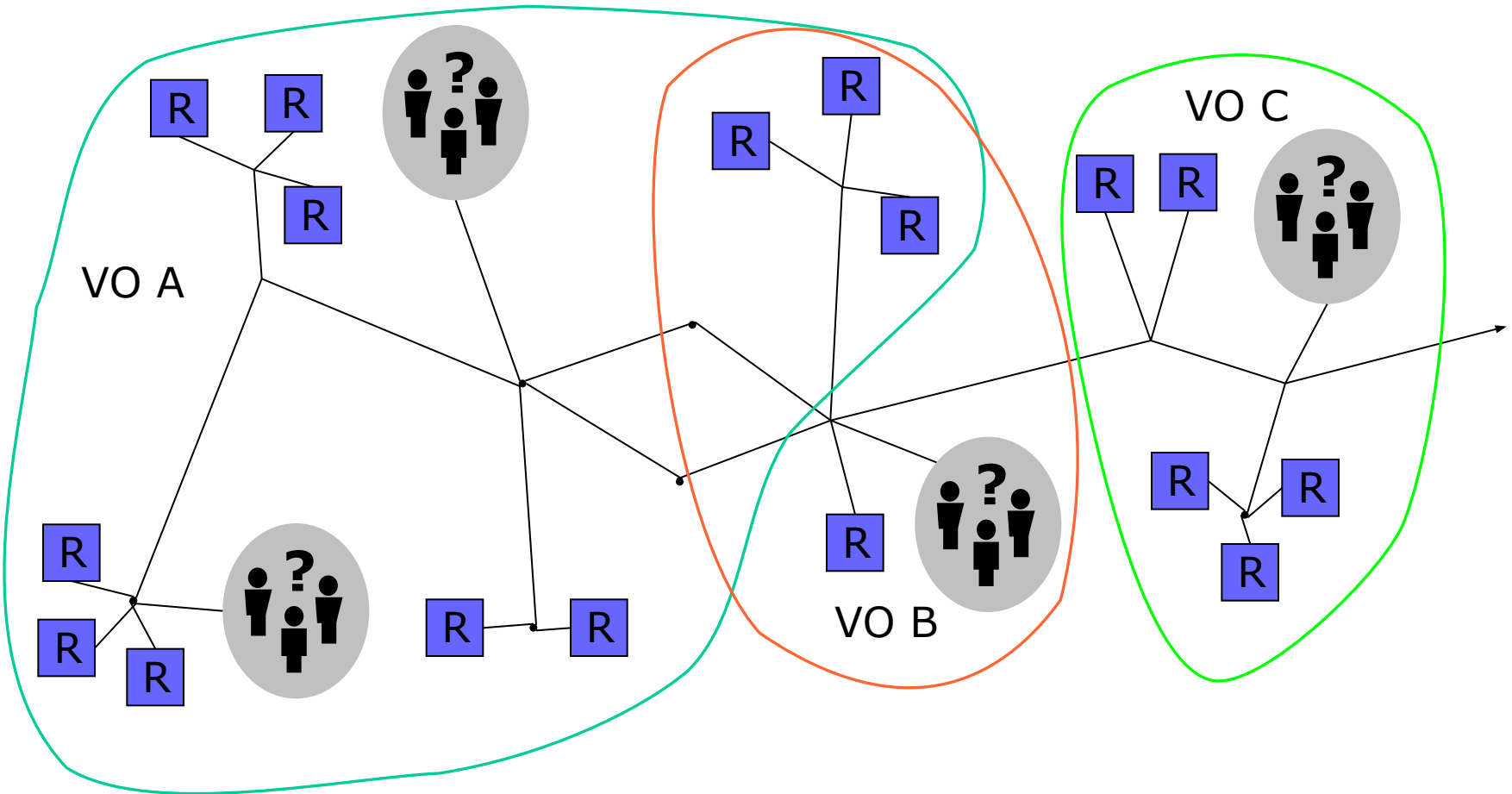


Grid Information Service

- Provide access to static and dynamic information regarding system components
- A basis for configuration and adaptation in heterogeneous, dynamic environments
- Requirements and characteristics
 - Uniform, flexible access to information
 - Scalable, efficient access to dynamic data
 - Access to multiple information sources
 - Decentralized maintenance



The GIS Problem: Many Information Sources, Many Views



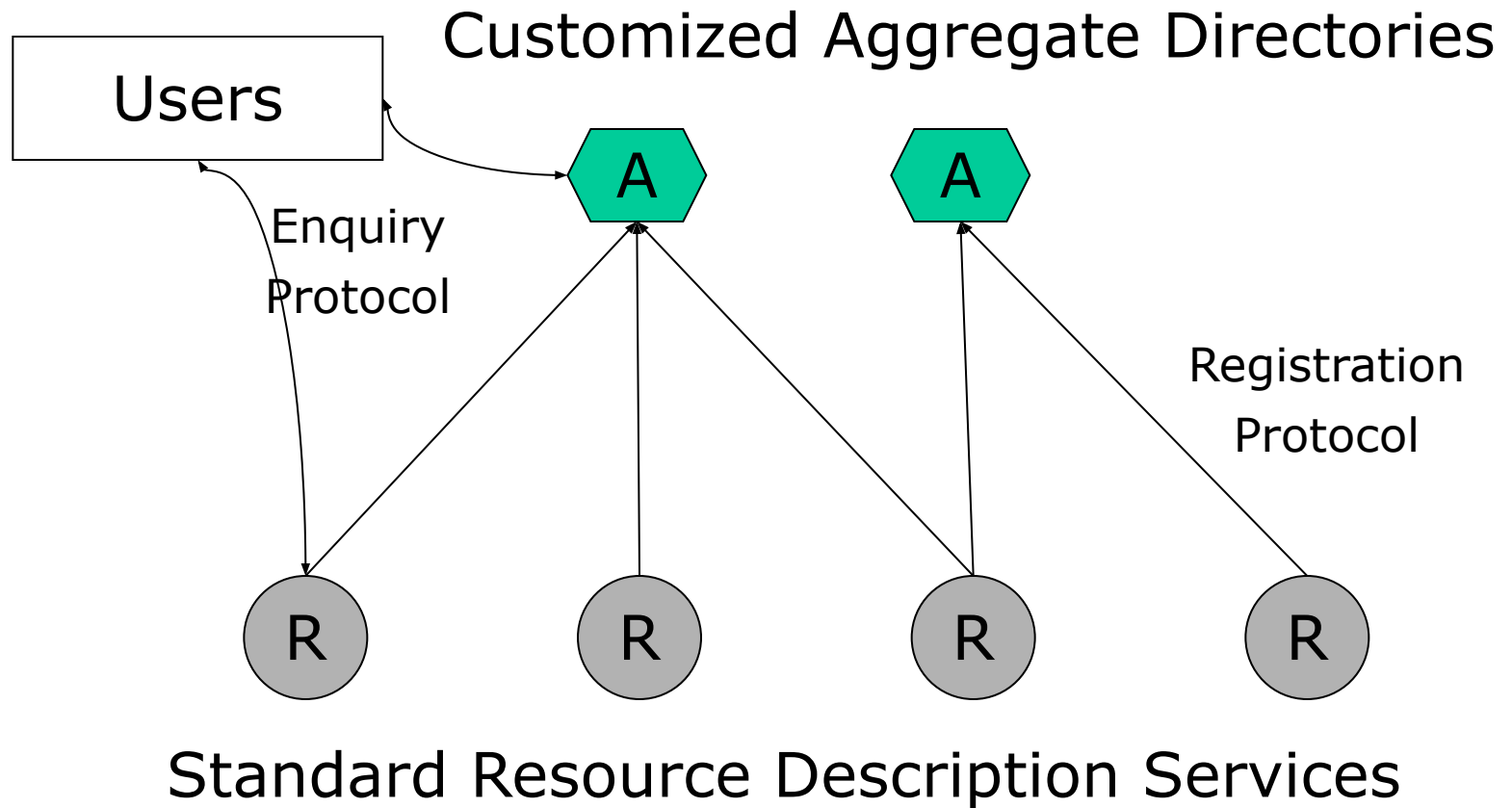


Information Protocols

- Grid Resource Registration Protocol
 - Support information/resource discovery
 - Designed to support machine/network failure
- Grid Resource Inquiry Protocol
 - Query resource description server for information
 - Query aggregate server for information
 - LDAP V3.0 in Globus 1.1.3



GIS Architecture





Metacomputing Directory Service

- Use LDAP as Inquiry
- Access information in a distributed directory
 - Directory represented by collection of LDAP servers
 - Each server optimized for particular function
- Directory can be updated by:
 - Information providers and tools
 - Applications (i.e., users)
 - Backend tools which generate info on demand
- Information dynamically available to tools and applications



LDAP Details

- Lightweight Directory Access Protocol
 - IETF Standard
 - Stripped down version of X.500 DAP protocol
 - Supports distributed storage/access (referrals)
 - Supports authentication and access control
- Defines:
 - Network protocol for accessing directory contents
 - Information model defining form of information
 - Namespace defining how information is referenced and organized



Information Services API

- RFC 1823 defines an IETF draft standard client API for accessing LDAP databases
 - Connect to server
 - Pose query which returns data structures contains sets of object classes and attributes
 - Functions to walk these data structures
- Globus does not provide an LDAP API. We recommend the use of OpenLDAP, an open source implementation of RFC 1823.



Searching an LDAP Directory

grid-info-search [options] filter [attributes]

- Default **grid-info-search** options

- h **mds.globus.org** *MDS server*
- p **389** *MDS port*
- b **"o=Grid"** *search start point*
- T **30** *LDAP query timeout*
- s **sub** *scope = subtree*

alternatives:

base : *lookup this entry*

one : *lookup immediate children*



The Globus Toolkit™:
Data Management
Services

The Globus Project™

Argonne National Laboratory
USC Information Sciences Institute

<http://www.globus.org>



Data Grid Problem

- “Enable a geographically distributed community [of thousands] to pool their resources in order to perform sophisticated, computationally intensive analyses on Petabytes of data”
- Note that this problem:
 - Is common to many areas of science
 - Overlaps strongly with other Grid problems



Major Data Grid Projects

- Earth System Grid (DOE Office of Science)
 - DG technologies, climate applications
- European Data Grid (EU)
 - DG technologies & deployment in EU
- GriPhyN (NSF ITR)
 - Investigation of “Virtual Data” concept
- Particle Physics Data Grid (DOE Science)
 - DG applications for HENP experiments



Data Intensive Issues Include ...

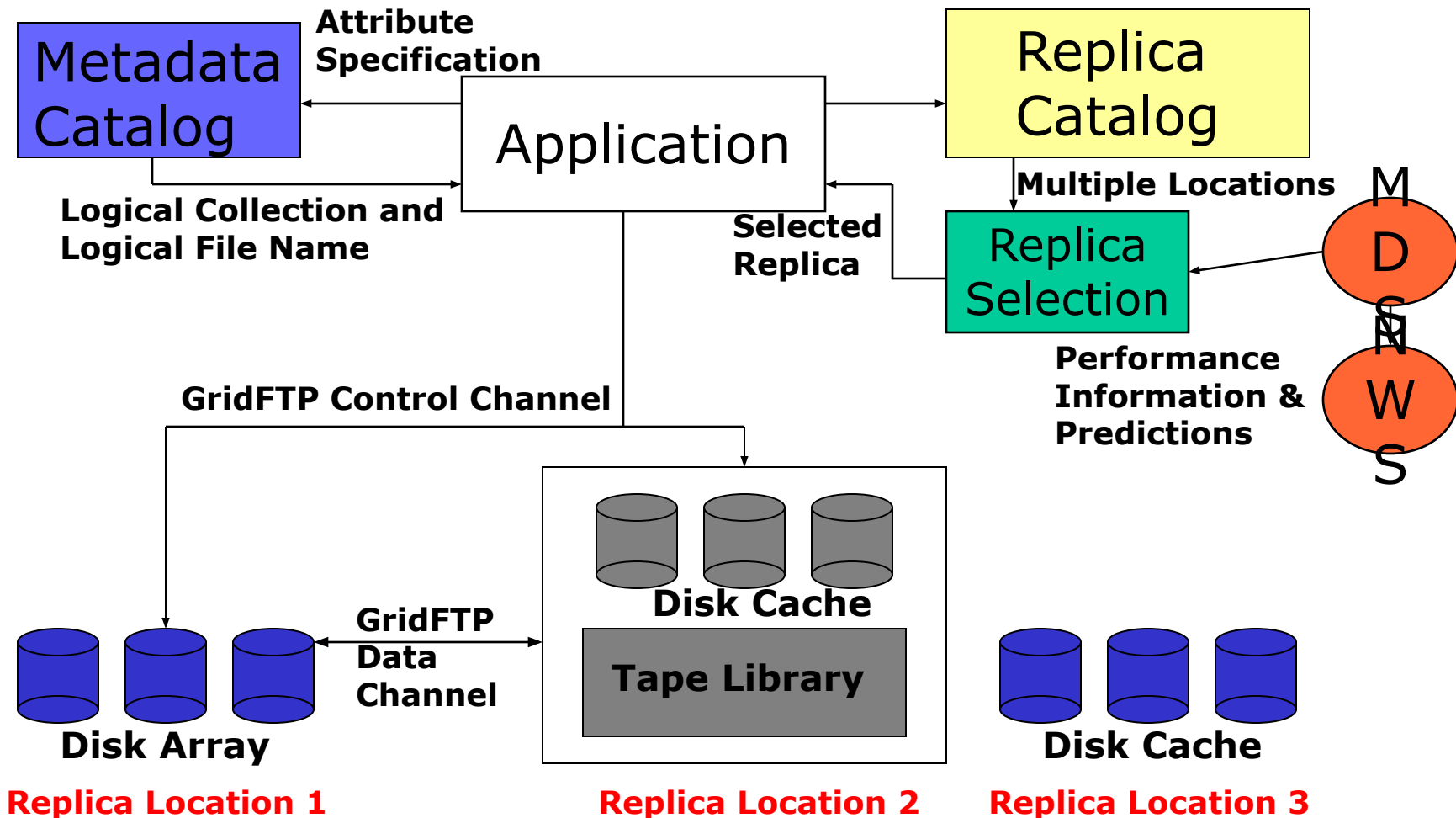
- Harness [potentially large numbers of] data, storage, network resources located in distinct administrative domains
- Respect local and global policies governing what can be used for what
- Schedule resources efficiently, again subject to local and global constraints
- Achieve high performance, with respect to both speed and reliability
- Catalog software and virtual data



Data Intensive Computing and Grids

- The term “Data Grid” is often used
 - Unfortunate as it implies a distinct infrastructure, which it isn’t; but easy to say
- Data-intensive computing shares numerous requirements with collaboration, instrumentation, computation, ...
 - Security, resource mgt, info services, etc.
- Important to exploit commonalities as very unlikely that multiple infrastructures can be maintained
- Fortunately this seems easy to do!

A Model Architecture for Data Grids





Globus Toolkit Components

Two major Data Grid components:

1. Data Transport and Access

- Common protocol
 - Secure, efficient, flexible, extensible data movement
- Family of tools supporting this protocol

2. Replica Management Architecture

- Simple scheme for managing:
 - multiple copies of files
 - collections of files



Access/Transport Protocol Requirements

- Suite of communication libraries and related tools that support
 - GSI, Kerberos security
 - Third-party transfers
 - Parameter set/negotiate
 - Partial file access
 - Reliability/restart
 - Large file support
 - Data channel reuse
 - Integrated instrumentation
 - Login/audit trail
 - Parallel transfers
 - Striping (cf DPSS)
 - Policy-based access control
 - Server-side computation
 - Proxies (firewall, load bal)
- All based on a standard, widely deployed protocol



And The Protocol Is ... GridFTP

- Why FTP?
 - Ubiquity enables interoperation with many commodity tools
 - Already supports many desired features, easily extended to support others
 - Well understood and supported
- We use the term GridFTP to refer to
 - Transfer protocol which meets requirements
 - Family of tools which implement the protocol
- Note GridFTP > FTP
- Note that despite name, GridFTP is not restricted to file transfer!



GridFTP: Basic Approach

- FTP protocol is defined by several IETF RFCs
- Start with most commonly used subset
 - Standard FTP: get/put etc., 3rd-party transfer
- Implement standard but often unused features
 - GSS binding, extended directory listing, simple restart
- Extend in various ways, while preserving interoperability with existing servers
 - Striped/parallel data channels, partial file, automatic & manual TCP buffer setting, progress monitoring, extended restart



GridFTP Protocol Specifications

- Existing standards
 - RFC 949: File Transfer Protocol
 - RFC 2228: FTP Security Extensions
 - RFC 2389: Feature Negotiation for the File Transfer Protocol
 - Draft: FTP Extensions
- New drafts
 - GridFTP: Protocol Extensions to FTP for the Grid
 - > Grid Forum Data Working Group



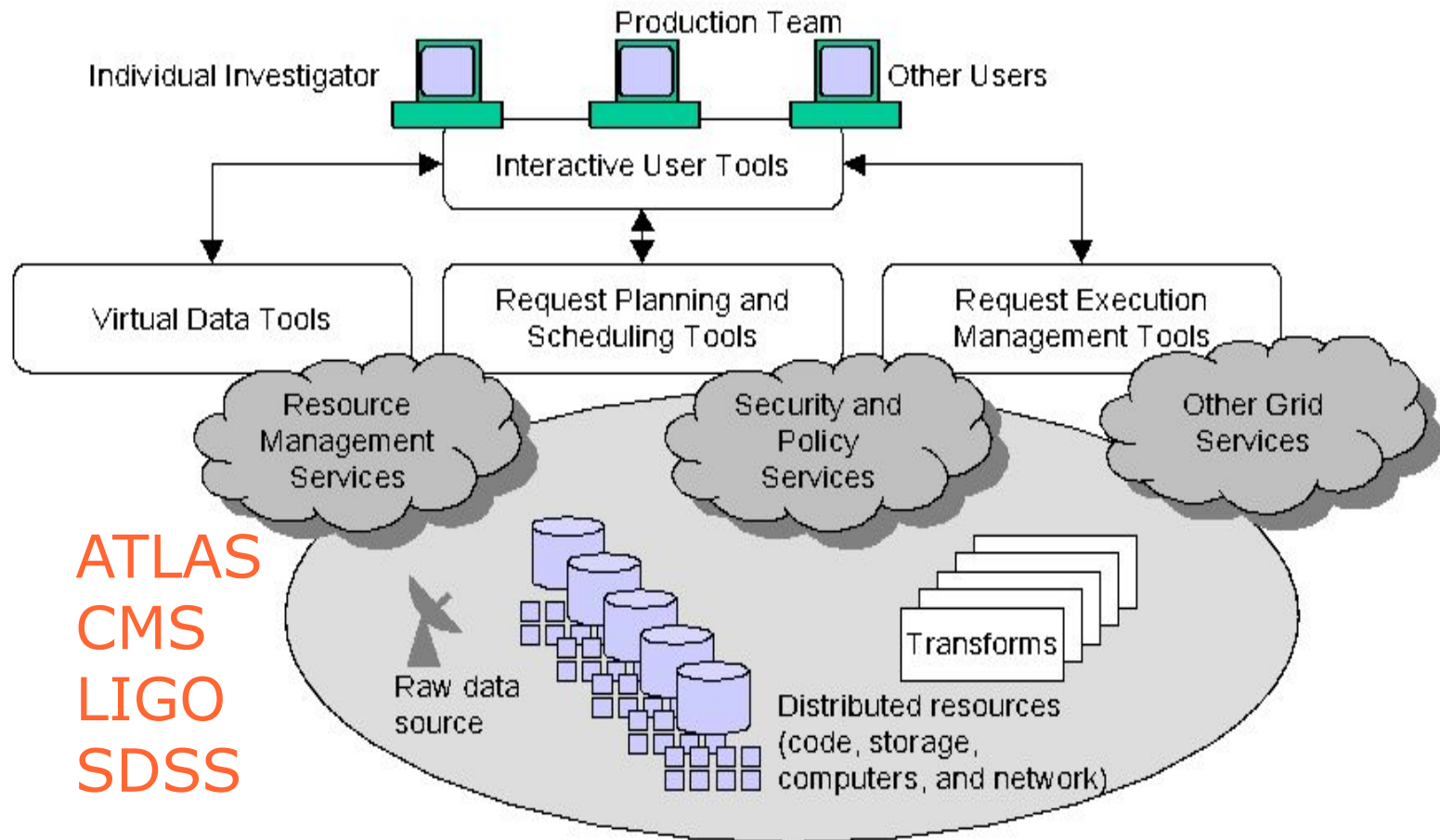
A Word on GASS

- The Globus Toolkit provides services for file and executable staging and I/O redirection that work well with GRAM. This is known as Globus Access to Secondary Storage (GASS).
- GASS uses GSI-enabled HTTP as the protocol for data transfer, and a caching algorithm for copying data when necessary.
- The `globus_gass`, `globus_gass_transfer`, and `globus_gass_cache` APIs provide programmer access to these capabilities, which are already integrated with the GRAM job submission tools.



Grid Physics Network (GriPhyN)

*Enabling R&D for advanced data grid systems,
focusing in particular on Virtual Data concept*





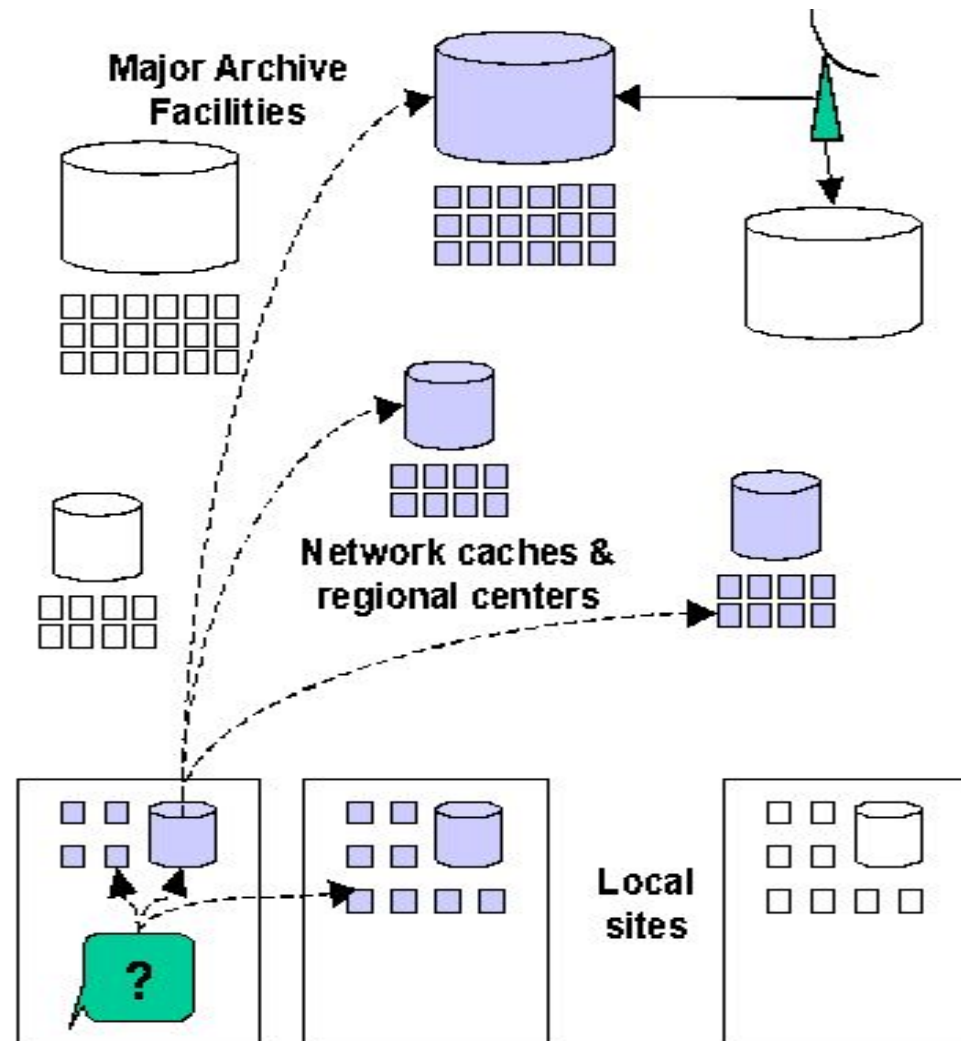
The Virtual Data Concept

“[a virtual data grid enables] the definition and delivery of a potentially unlimited virtual space of data products derived from other data. In this virtual space, requests can be satisfied via direct retrieval of materialized products and/or computation, with local and global resource management, policy, and security constraints determining the strategy used.”



Virtual Data in Action

- Data request may
 - Access local data
 - Compute locally
 - Compute remotely
 - Access remote data
- Scheduling subject to local & global policies
- Local autonomy





Futures & Conclusions

The Globus Project™

Argonne National Laboratory

USC Information Sciences Institute

<http://www.globus.org>



Problem Evolution

- Past-present: $O(10^2)$ high-end systems; Mb/s networks; centralized (or entirely local) control
 - I-WAY (1995): 17 sites, week-long; 155 Mb/s
 - GUSTO (1998): 80 sites, long-term experiment
 - NASA IPG, NSF NTG: $O(10)$ sites, production
- Present: $O(10^4-10^6)$ data systems, computers; Gb/s networks; scaling, decentralized control
 - Scalable resource discovery; restricted delegation; community policy; Data Grid: 100s of sites, $O(10^4)$ computers; complex policies
- Future: $O(10^6-10^9)$ data, sensors, computers; Tb/s networks; highly flexible policy, control



The Future: All Software is Network-Centric

- We don't build or buy "computers" anymore, we borrow or lease required resources
 - When I walk into a room, need to solve a problem, need to communicate
- A "computer" is a dynamically, often collaboratively constructed collection of processors, data sources, sensors, networks
 - Similar observations apply for software



And Thus ...

- Reduced barriers to access mean that we do much more computing, and more interesting computing, than today => Many more components (& services); massive parallelism
- All resources are owned by others => Sharing (for fun or profit) is fundamental; trust, policy, negotiation, payment
- All computing is performed on unfamiliar systems => Dynamic behaviors, discovery, adaptivity, failure



Summary

- The Grid problem: Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations
- Grid architecture: Emphasize protocol and service definition to enable interoperability and resource sharing
- Globus Toolkit™ a source of protocol and API definitions, reference implementations
- See: www.globus.org, www.gridforum.org



Вообще-то говоря, это
последний слайд...

The Globus Project™

Argonne National Laboratory
USC Information Sciences Institute

<http://www.globus.org>