



Open Source Grid and Cluster Conference 2008

Globus Primer: Introduction to Globus Software

Lee Liming

University of Chicago • Argonne National Laboratory



THE UNIVERSITY OF
CHICAGO



Approach

- I. What can Globus do for me?
- II. Where does Globus software help in a Grid system or application?
- III. Using Globus to locate services
- IV. Using Globus to share data
- V. Using Globus for massive data analysis
- VI. Using Globus to scale an application

For final slides (minor updates, in color) go to
<http://www.mcs.anl.gov/~liming/primer/>

What Can Globus Do For Me?



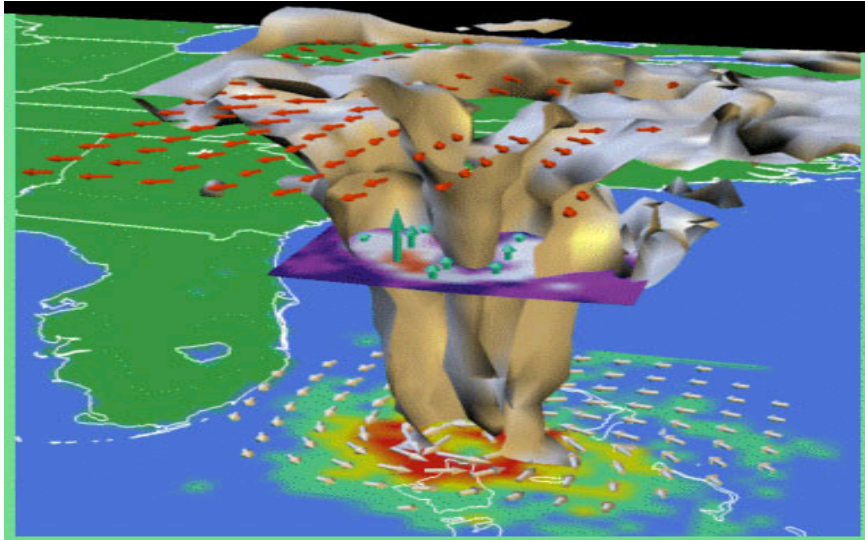
THE UNIVERSITY OF
CHICAGO



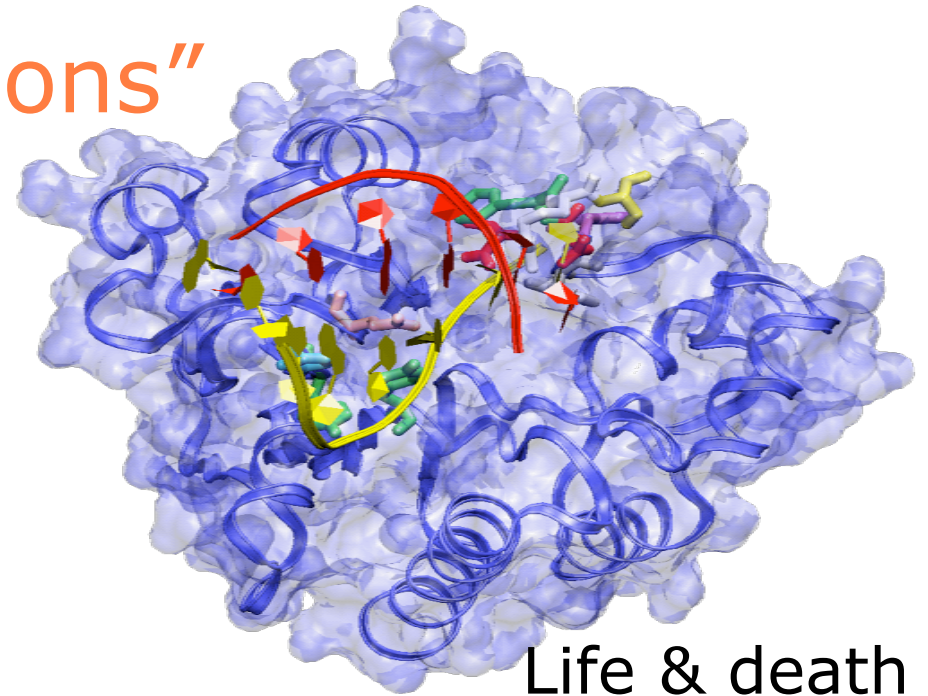
A Word on Context

- Globus arose from issues in science
 - ◆ Requirements came from (often big) science enterprises
 - ◆ Majority of Globus developers are (still) funded by and focused on science needs
 - ◆ This presentation is focused on science examples
- If your interest is in how Globus applies to business needs, this *should* bother you a little (but not a lot)
 - ◆ The same was true of other things that have created huge business opportunities: mathematics, engineering, computers, internet, web, MRI, the space program...
 - ◆ The avenues to business needs are there, but they aren't marked well and they haven't been paved
 - ◆ We (reasonably) expect that business use of Globus will catch up with and surpass scientific/engineering use, but it hasn't yet

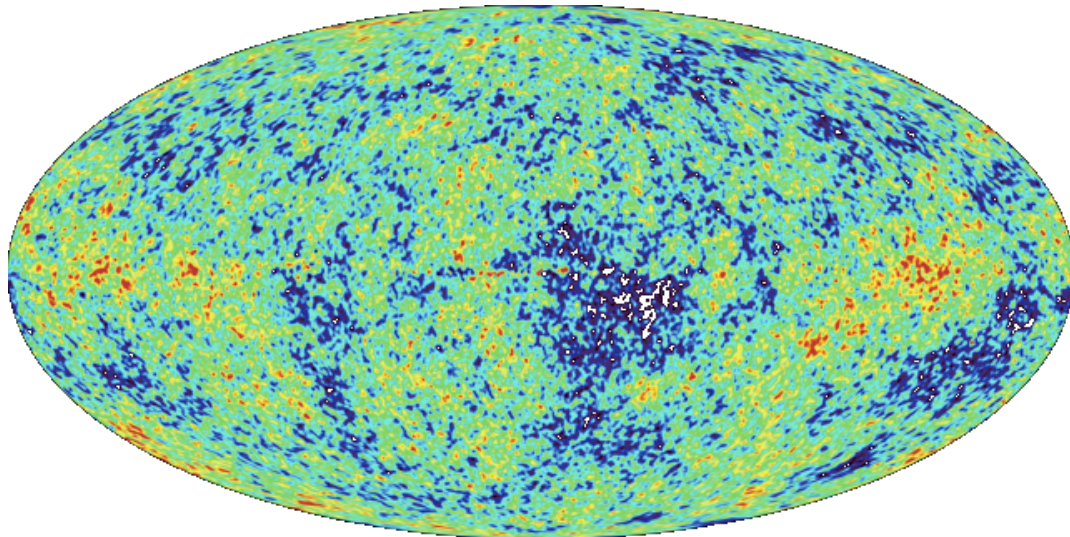
Science's "Big Questions"



Future of the planet



Life & death



Nature of the universe



Consciousness

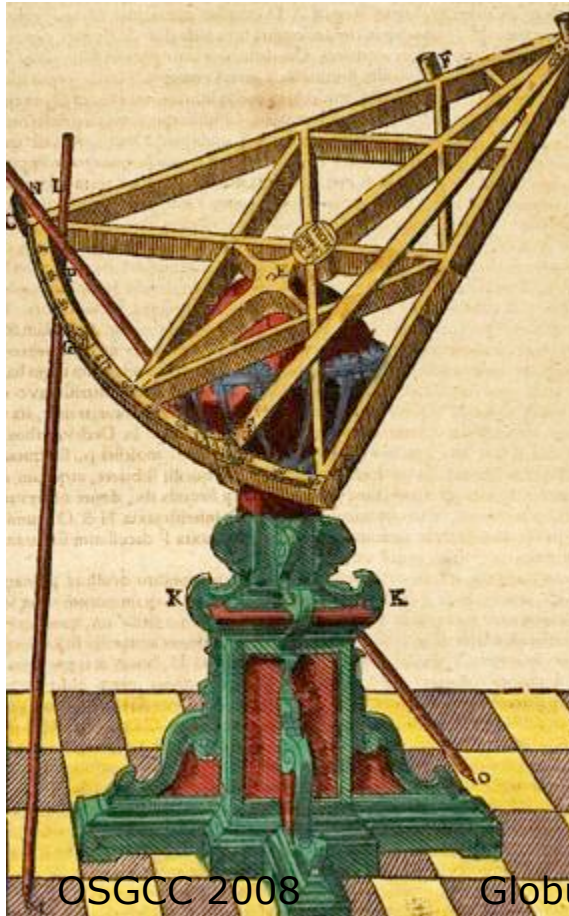
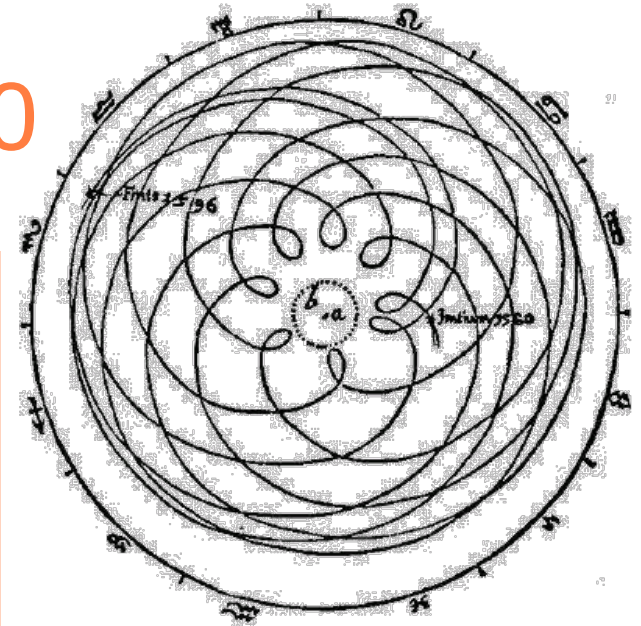


the globus alliance

www.globus.org

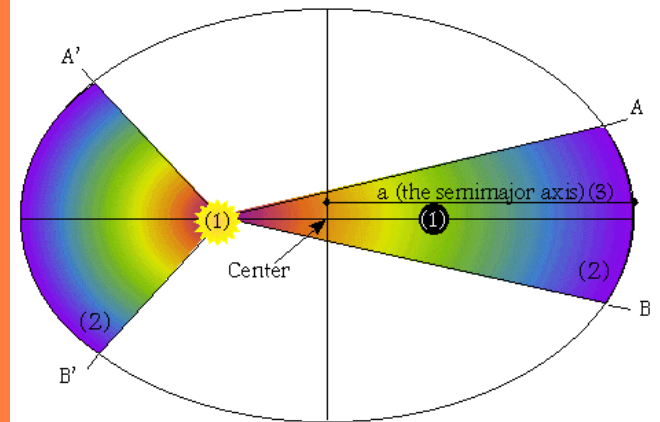
Scientific Communication, ~1600

DE MOTIB. STELLÆ MARTIS



Brahe

Kepler







Scientific Communication, ~2000 Service-Oriented Science

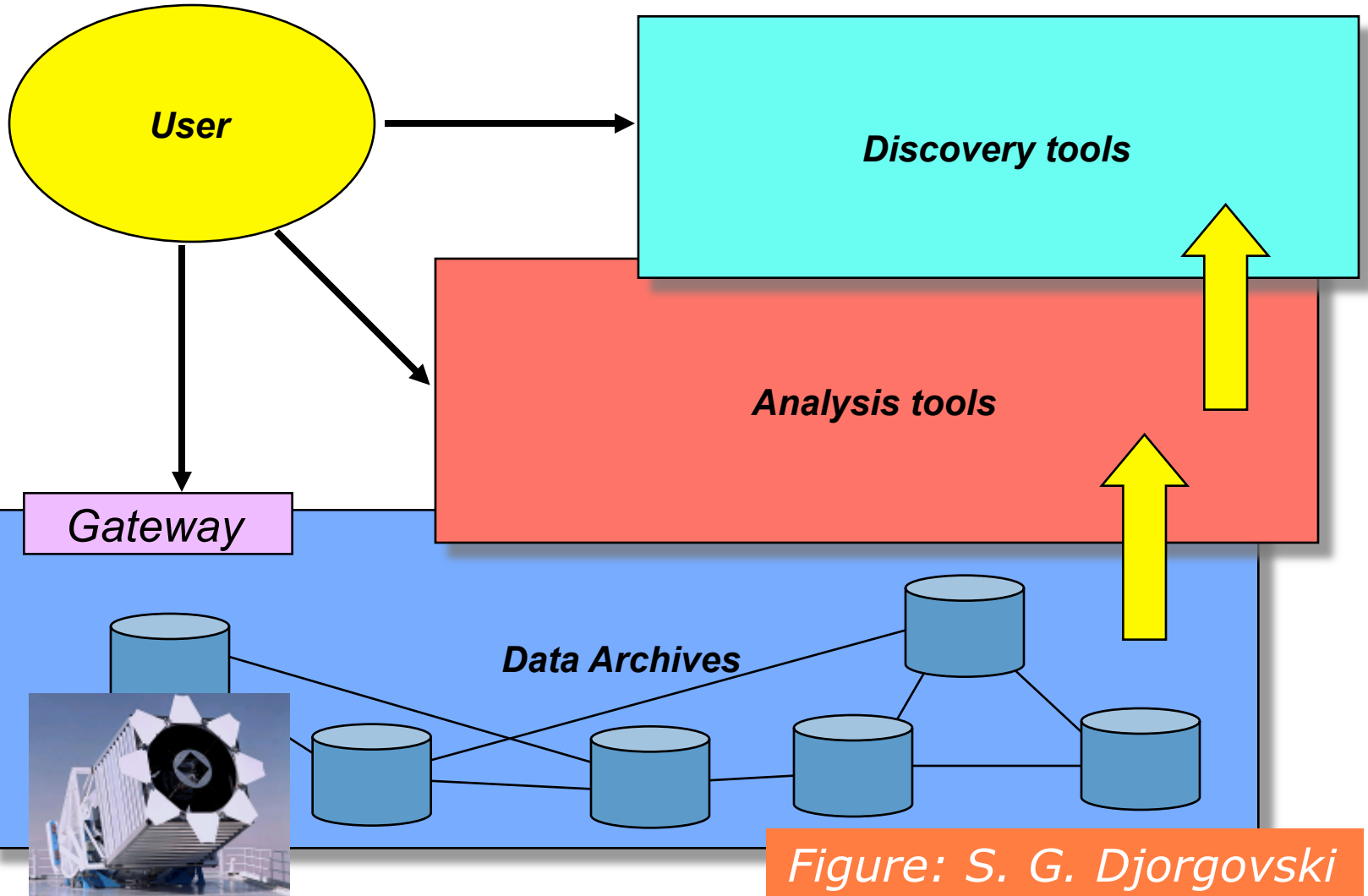
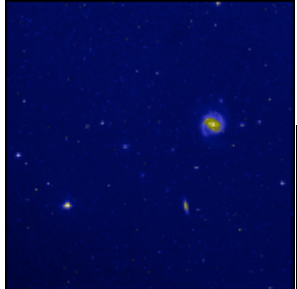
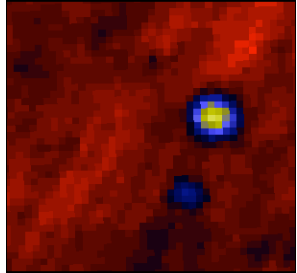
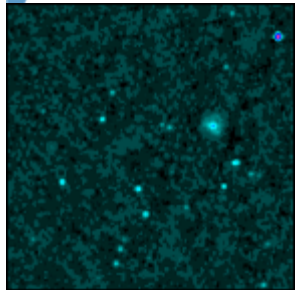


Figure: S. G. Djorgovski



What are the Products of Science?

- Papers
 - ◆ "We learned this, and here's how."
- Data and datasets
 - ◆ "We collected this data. Download it, write it up, and see what you can learn from it."
- Web portals
 - ◆ "We constructed this scientific model. Use it on your own, supply some parameters, and see how it works."
 - ◆ Requires manual operation.
- Web services
 - ◆ "Here's our climate model. Integrate it with your model for [ocean currents/weather/crop forecasts] and see what happens."
 - ◆ "Here's our indexed data from the latest experiments. Filter your filters against it and see if you can find anything interesting."
 - ◆ "Here's our genome analysis engine. Upload your proteins and see what they will do in a cell."

Increasing
degrees of
collaboration



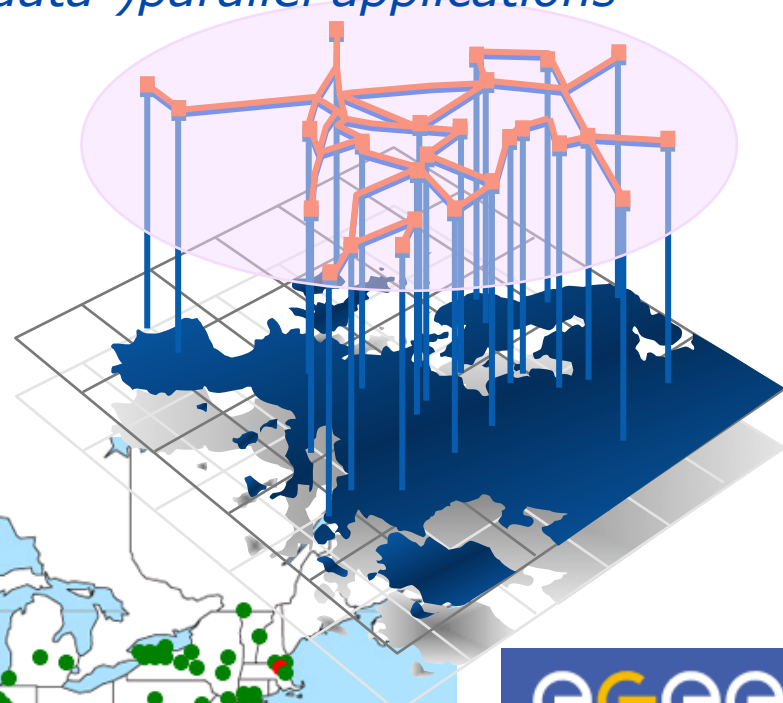
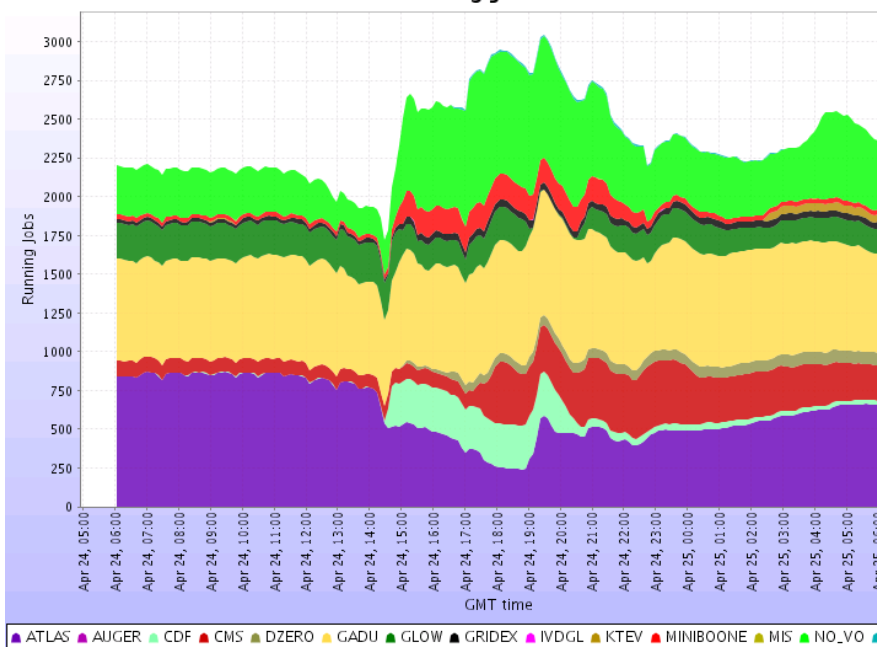
the globus alliance

www.globus.org

1st-Generation Grids: On-Demand/Batch Computing

Focus on resource aggregation for (data-)parallel applications

Running Jobs



Open Science Grid



OSGCC 2008

Globus Primer: An Introduction to Globus Software



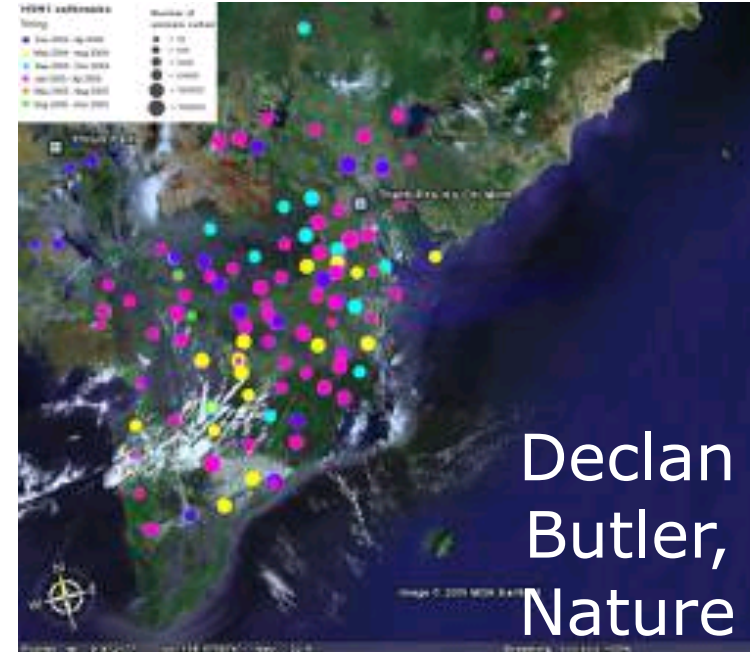


the globus alliance

www.globus.org

"Web 2.0"

- Software as services
 - ◆ Data- & computation-rich network services
- Services as platforms
 - ◆ Easy composition of services to create new capabilities ("mashups")—that themselves may be made accessible as new services
- Enabled by massive infrastructure buildout
 - ◆ Google spent \$1B+ on computers, networks, and real estate in 2006
 - ◆ Many others are spending substantially
- Paid for by advertising





the globus alliance

www.globus.org

2nd-Generation Grids: Service-Oriented Science

- Empower many more users by enabling on-demand access to **services**
- Grids become an enabling technology for **service-oriented science** (and business)
 - ◆ Grid infrastructures *host* services
 - ◆ Grid applications *use* (and *are*) services
 - ◆ Grid development tools *build* services



TeraGrid™
EMPOWERING DISCOVERY

Science
Gateways



"Service-Oriented Science", Science, 2005



Automating Science

- Human access to data is nice
- Automated access by software tools is revolutionary

In the time that a human user takes to locate one useful piece of information within a Web site, a program may access and integrate data from many sources and identify relationships that a human might never discover unaided. (Foster)



Service-Oriented Science

People **create** services (data or functions) ...
which I **discover** (& decide whether to use) ...
& **compose** to create a new function ...
& then **publish** as a new service.

→ I find "someone else" to **host** services,
so I don't have to become an expert in operati
services & computers!



→ I hope that this "someone else" can
manage security, reliability, scalability, ...



TeraGrid[™]
EMPOWERING DISCOVERY





(Web) Service-Oriented Business

My company (or a partner or potential partner) **creates** components (product subsystems) ...
which I **discover** (& decide whether to use) ...
& **compose** to create a new product...
& then **host** for delivery to customers.

→ *I find "someone else" to **host** services,
so I don't have to become an expert in operating
services & computers!*



→ *I hope that this "someone else" can
manage security, reliability, scalability, ...*





A 20,000-foot-high Answer

- Globus provides building blocks for constructing service-oriented systems and applications
 - ◆ Often needed by *virtual organizations*
 - ◆ A new way to do things, enabled by information technology



Globus Philosophy

- Globus was first established as an open source project in 1996
- The Globus Toolkit is open source to:
 - ◆ Allow for inspection
 - for consideration in standardization processes
 - ◆ Encourage adoption
 - in pursuit of ubiquity and interoperability
 - ◆ Encourage contributions
 - harness the expertise of the community
- The Globus Toolkit is distributed under the (BSD-style) Apache license version 2



dev.globus

- Governance model based on Apache Jakarta
 - ◆ Consensus based decision making
- Globus software is organized as several dozen *Globus projects*
 - ◆ Each project has its own *committers* responsible for their products
 - ◆ Cross-project coordination through shared interactions and committers meetings
- Globus management committee (GMC)
 - ◆ Overall guidance and conflict resolution



Guidelines
(Apache
Jakarta)

Infrastructure
(CVS, email,
bugzilla, Wiki)

Projects
Include

...

- Welcome
- List of projects
- Guidelines
- Infrastructure
- How to contribute
- GlobDev events
- Recent changes
- GlobDev FAQ

- common runtime projects*
- C Core Utilities
 - C WS Core
 - CoG jglobus
 - Core WS Schema
 - Java WS Core
 - Python Core
 - XIO

- data projects*
- GridFTP
 - OGSA-DAI
 - Reliable File Transfer
 - Replica Location

- execution projects*
- GRAM

- information projects*
- MDS4

- security projects*
- C Security
 - CAS/SAML Utilities
 - Delegation Service

Welcome

This is the new home Globus software development; it is still under construction. The current status of our efforts to build this environment can be found [on this page](#). Comments regarding this site can be sent to info@globus.org. Thank you for your interest in Globus development!

Globus was first established as an open source software project in 1996. Since that time, the Globus development team has expanded from a few individuals to a distributed, international community. In response to this growth, the Globus community (the "Globus Alliance") established in October 2005 a new source code development *infrastructure* and meritocratic *governance model*, which together make the process by which a developer joins the Globus community both easier and more transparent.

The Globus governance model and infrastructure are based on those of [Apache Jakarta](#). In brief, the governance model places control over each individual software component (*project*) in the hands of its most active and respected *contributors* (*committers*), with a *Globus Management Committee* (GMC) providing overall guidance and conflict resolution. The infrastructure comprises *repositories*, *email lists*, Wikis, and *bug trackers* configured to support per-project community access and management.

For more information, see:

- The [Globus Alliance Guidelines](#), which address various aspects of the Globus governance model and the Globus community.
- A description of the Globus Alliance [Infrastructure](#).
- A list of current Globus projects.
- Information about Globus community events.
- The [conventions and guidelines](#) that apply to contributions.



Open Source != "Free time"

- **Globus development is well-funded**
 - ◆ The open source model facilitates contributions
 - ◆ NSF and DOE sponsor Globus development at several institutions via multiple grants, totaling >\$5M/yr
 - ◆ Non-U.S. science agencies also contribute to Globus development
 - ◆ Corporations also sponsor developers
- **NSF explicitly funds Globus improvements**
 - ◆ CDIGS: Community-Driven Improvements to Globus Software



Globus Technology Areas

- Core runtime
 - ◆ Infrastructure for building new services
- Security
 - ◆ Apply uniform policy across distinct systems
- Execution management
 - ◆ Provision, deploy, & manage services
- Data management
 - ◆ Discover, transfer, & access large data
- Monitoring
 - ◆ Discover & monitor dynamic services



Globus Software: dev.globus.org

Globus Projects

MPICH G2

OGSA-DAI

Incubation
Mgmt

Java
Runtime

Delegation

MyProxy

GRAM

Data
Rep

Replica
Location

C
Runtime

CAS

GSI-
OpenSSH

GridFTP

MDS4

Python
Runtime

C Sec

GridWay

Reliable
File
Transfer

GT4 Docs

Globus Toolkit

Incubator Projects

Swift

GEMLCA

gRAVI

MonMan

GAARDS

MEDICUS

Cog WF

Virt WkSp

NetLogger

GDTE

GridShib

OGRO

UGP

Dyn Acct

Gavia JSC

DDM

Metrics

Introduce

PURSE

HOC-SA

LRMA

WEEP

Gavia MS

SGGC

ServMark

**Common
Runtime**

Security

**Execution
Mgmt**

Data Mgmt

**Info
Services**

Other



What Is the Globus Toolkit?

- The Globus Toolkit is a collection of solutions to problems that frequently come up when trying to build collaborative distributed applications
- Heterogeneity
 - ◆ To date (v1.0 - v4.0), the Toolkit has focused on *simplifying heterogeneity* for application developers
 - ◆ We are increasingly including more “vertical solutions” that implement typical application patterns
- Security
 - ◆ The Grid Security Infrastructure (GSI) allows collaborators to share resources without blind trust
- Standards
 - ◆ Our goal has been to capitalize on and encourage use of existing standards (IETF, W3C, OASIS, GGF)
 - ◆ The Toolkit also includes reference implementations of new/proposed standards in these organizations



What's In the Globus Toolkit?

- A Grid development environment
 - ◆ Develop new OGSA-compliant Web Services
 - ◆ Develop applications using Java or C/C++ Grid APIs
 - ◆ Secure applications using basic security mechanisms
- A set of basic Grid services
 - ◆ Job submission/management
 - ◆ File transfer (individual, queued)
 - ◆ Database access
 - ◆ Data management (replication, metadata)
 - ◆ Monitoring/Indexing system information
- Tools and examples
- The prerequisites for many Grid community tools



Another Word on Context

- You might be ready for Globus if you are...
 - ◆ ...extending the limits of what has been done before
 - ◆ ...so fed up with trying to make IT do things it wasn't designed to do that you are willing to explore new territory
 - ◆ ...looking for something that will redraw the map
 - ◆ ...highly dependent on sharing data and other resources with people outside your organization
 - ◆ ...generally enthusiastic about trying new things
 - ◆ ...able to identify and fill in some missing pieces
- **Warning: Globus is not a "whole product"**
 - ◆ Many people are using Globus components to build some very interesting new applications! Will you?

Where Does Globus Software Help in a Grid System or Application?



THE UNIVERSITY OF
CHICAGO



Known Territories

- There are several ways that Globus software has been used that are well-trod paths
 - ◆ General-purpose Grid infrastructures
 - ◆ Domain-specific Grid infrastructures
 - ◆ Domain-specific Grid-enabled applications
 - ◆ General-purpose Grid-enabled application frameworks
- For each of these, there are many examples and lessons learned



General-purpose Grid Infrastructure

- Builds on existing systems
 - ◆ Clusters, mass storage, SMPs...
- Adds Globus services
 - ◆ GRAM, GridFTP, MDS, RLS, RFT...
- Offers users a service-based interaction model for general-purpose shared systems
 - ◆ Old: Terminal login interface
 - ◆ New: Web service interfaces
- Users need documentation, client toolkit or SDK, registration mechanism



Example Uses

- Replace id/password login system with single sign-on
- Register details about each host in a Web service registry
- Enable remote job submission
 - ◆ End user clients
 - ◆ “Science gateways” (portal clients)
 - ◆ Workflow engine clients
- Provide high-performance data transfer
- Provide managed data transfer



the globus alliance
www.globus.org

Examples of Production Scientific Grids

- APAC (Australia)
- China Grid
- China National Grid
- DGrid (Germany)
- EGEE
- NAREGI (Japan)
- Open Science Grid
- Taiwan Grid
- TeraGrid
- ThaiGrid
- UK Nat'l Grid Service





Domain-specific Grid Infrastructure

- Builds on existing tools
 - ◆ Databases, instruments, applications
- Adds Web service interfaces
 - ◆ Teleinstrumentation service, database access services, metadata and registry services, collaboration interfaces, portal
- Offers users a Web(-service)-based system for doing domain-specific work; e.g., automation, sharing, pre-configuration
- Users need documentation, system integration notes



Example Uses

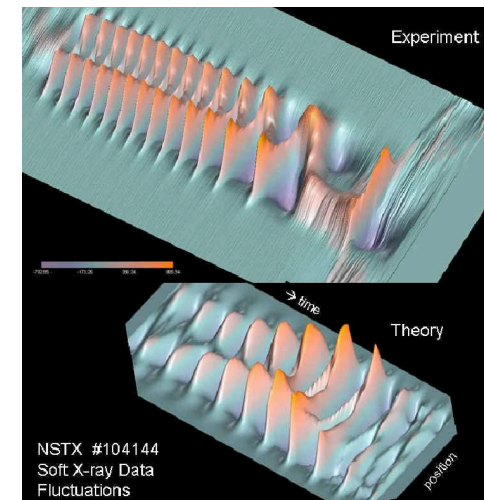
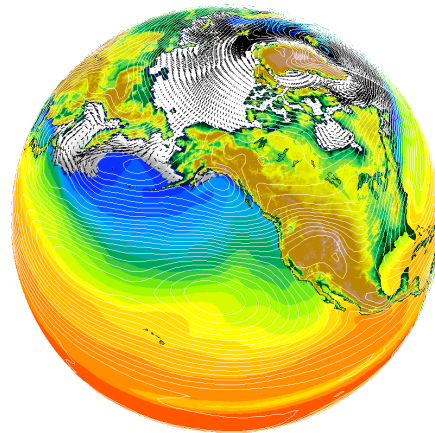
- Enable broader sharing of data, instruments, code within a community
- Lower the startup cost for new users of the community's data, instruments, code
- Build a community data repository
- Build an analysis/annotation community (Web 2.0 for science)



the globus alliance
www.globus.org

Examples of Domain-specific Scientific Grids

- caGrid – cancer research
- Earth System Grid – climate studies
- LEAD portal – weather forecasting
- LIGO – cosmology/physics
- NEES – earthquake engineering





Domain-specific Grid Applications

- Builds on existing application code
- Adds Web service interfaces
- Offers scaling and automation benefits
 - ◆ Can use (execute on) more systems
 - ◆ Can gather and use data from more places
 - ◆ Can automate repetitive tasks (workflow, fault recovery)
- Users need notes on how to use the application in a Grid setting, authorization to run on Grid systems



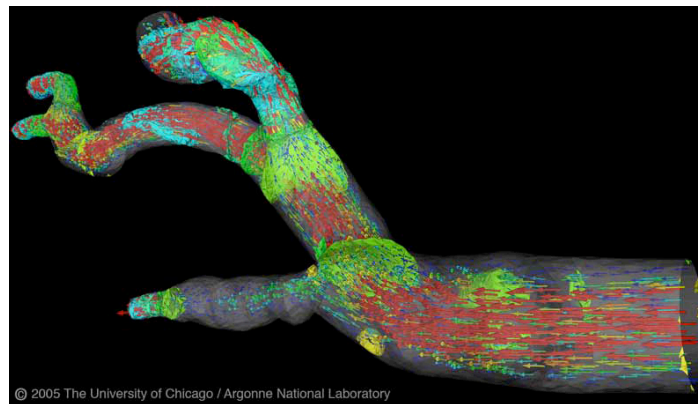
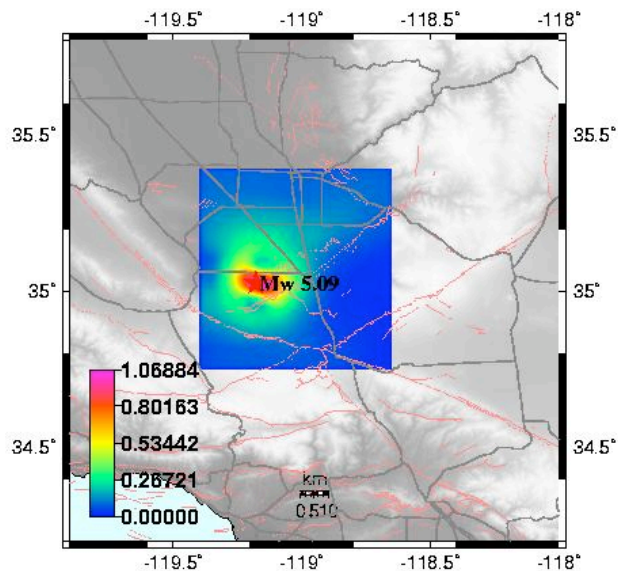
Example Uses

- Enable larger simulations (higher resolution, more realistic, larger area, linked models)
- Improve time-to-solution (massive parallelization, dynamic provisioning)
- Import data from more sources (discovery and data movement)
- Improve efficiency (dynamic provisioning)



Examples of Domain-specific Grid Applications

- LEAD portal – weather forecasting
- PUMA2 – functional genomics
- fMRI – functional MRI analysis
- TeraShake – seismological simulation





Gen-Purpose Grid App. Frameworks

- Builds on existing frameworks
 - ◆ MPI, RPC, Condor, etc.
- Adds support for Grid elements
 - ◆ GSI security, GRAM submission, GridFTP data movement, MDS resource discovery, etc.
- Offers application developers a familiar development framework with new Grid features, mostly for scaling up to multisystem runs
- Application developers need porting notes (porting to the Grid version of the framework)
- Users need release notes on any Grid configuration, authorization to run on Grid systems



Example Uses

- Simplify work of application developers in Grid-enabling their applications



Examples of Gen-purpose Grid App. Frameworks

- MPIg – MPI applications
- NinfG – RPC applications
- NetSolve – RPC applications
- Condor-G – Embarrassingly parallel and workflow applications
- MyCluster – Embarrassingly parallel and workflow applications



Current Challenges

- Application and Service “Hosting”
 - ◆ Service interfaces for legacy applications
 - ◆ Automated hosting/provisioning services
- Composition and Workflow
 - ◆ “CAD for applications”
 - ◆ Discovery and metadata
 - ◆ Automating the mundane

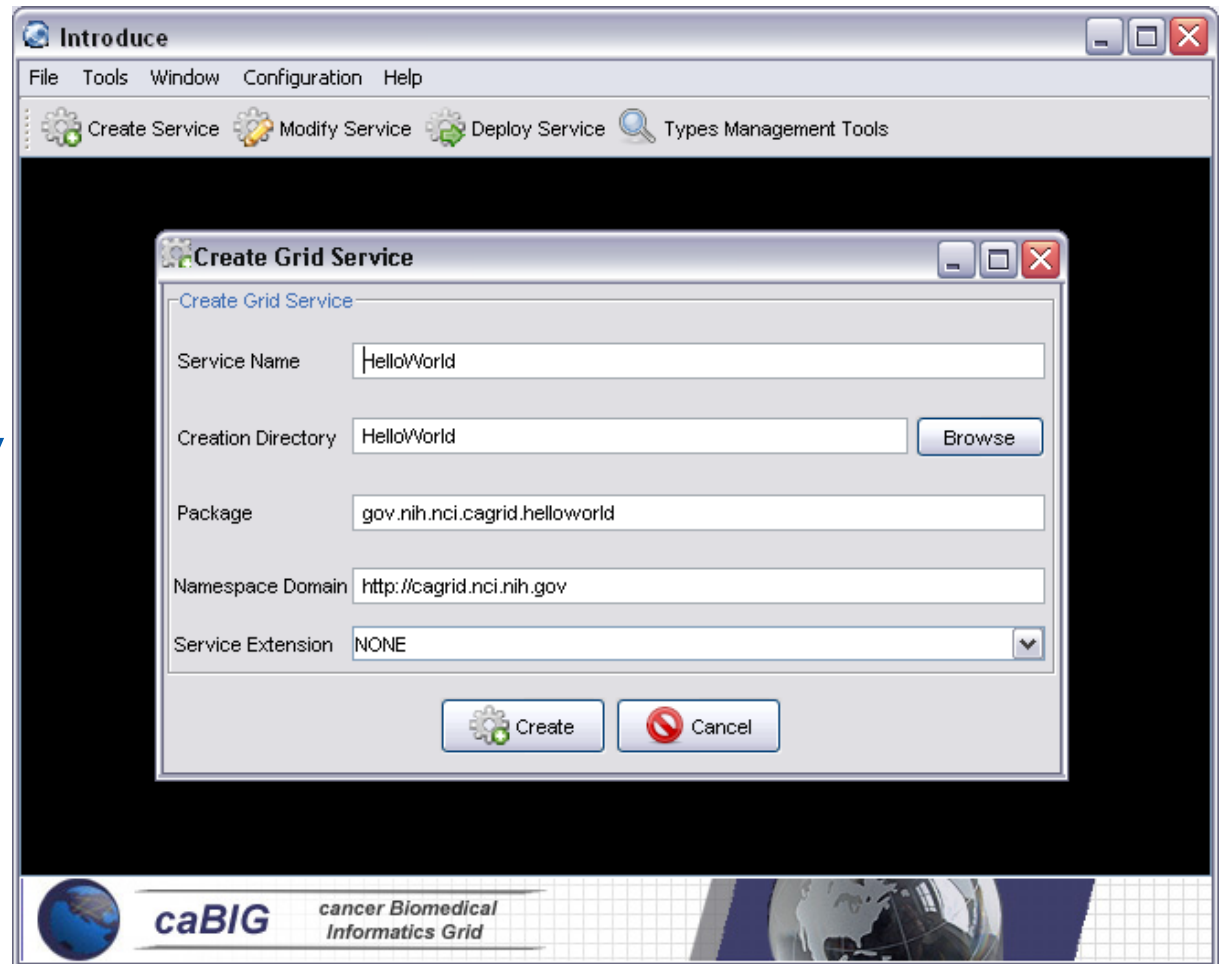


the globus alliance
www.globus.org

The Introduce Authoring Tool

- Define service
- Create skeleton
- Discover types
- Add operations
- Configure security
- Modify service

Generates GT4-compatible Web Services



Introduce: Hastings, Saltz, et al., Ohio State University



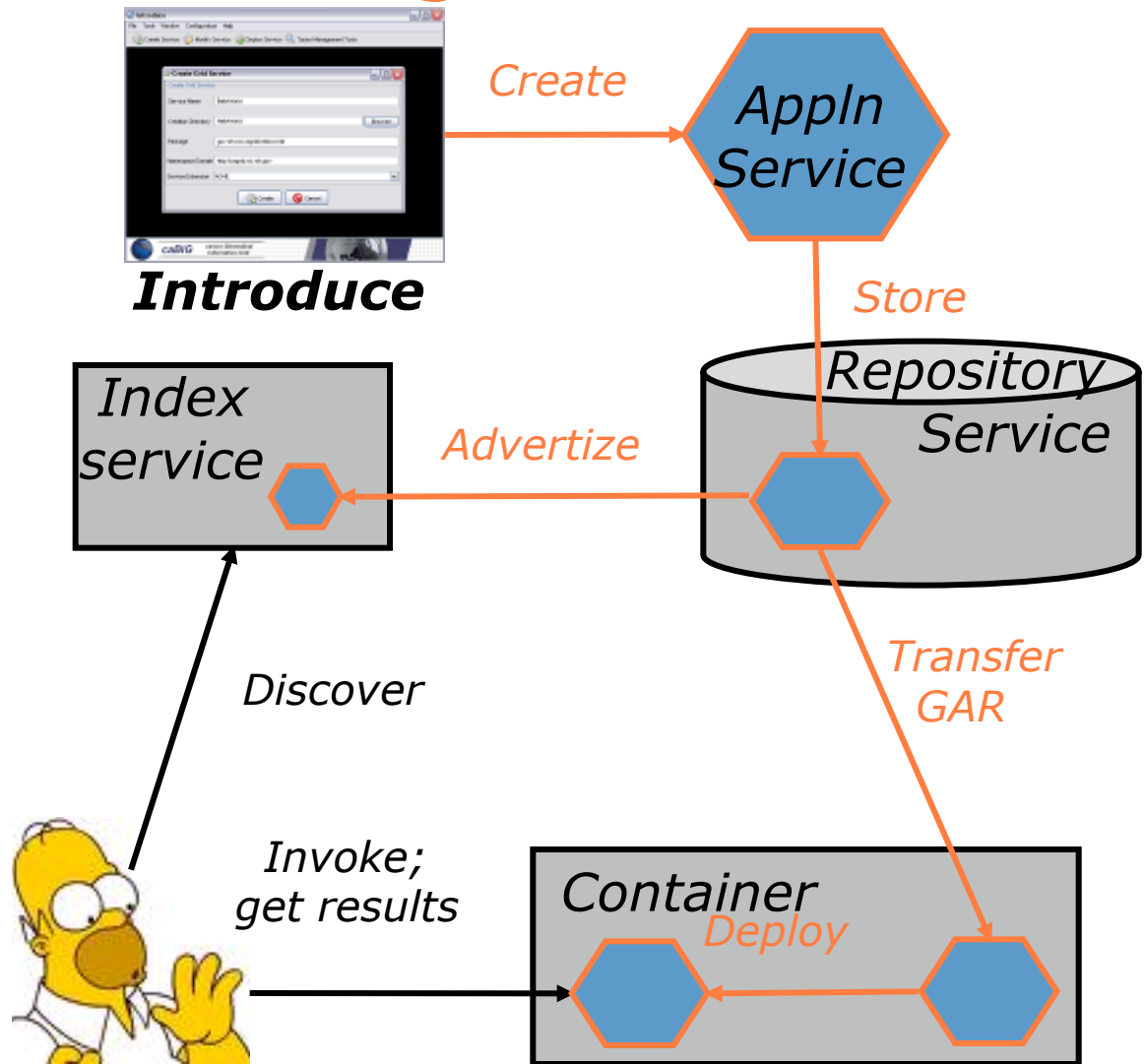
Introduce and gRAVI

- Introduce

- ◆ Define service
- ◆ Create skeleton
- ◆ Discover types
- ◆ Add operations
- ◆ Configure security

- **Grid Remote Application Virtualization Infrastructure**

- ◆ Wrap executables

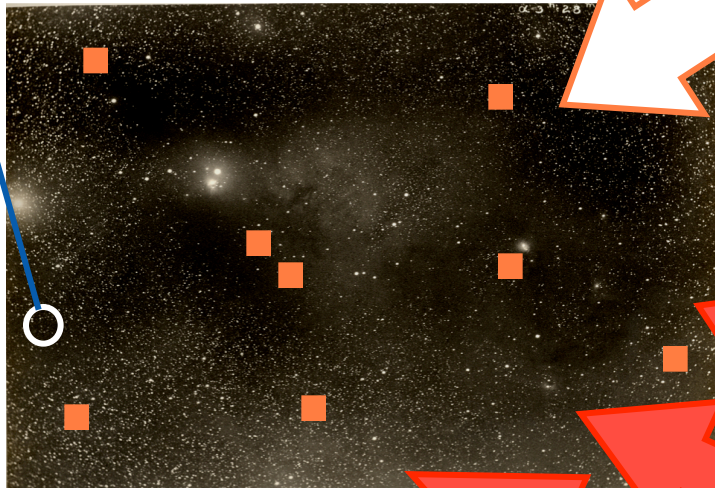


Argonne/U.Chicago & Ohio State University

The Importance of "Hosting" and "Management"

Tell me about this star

Tell me about these 20K stars



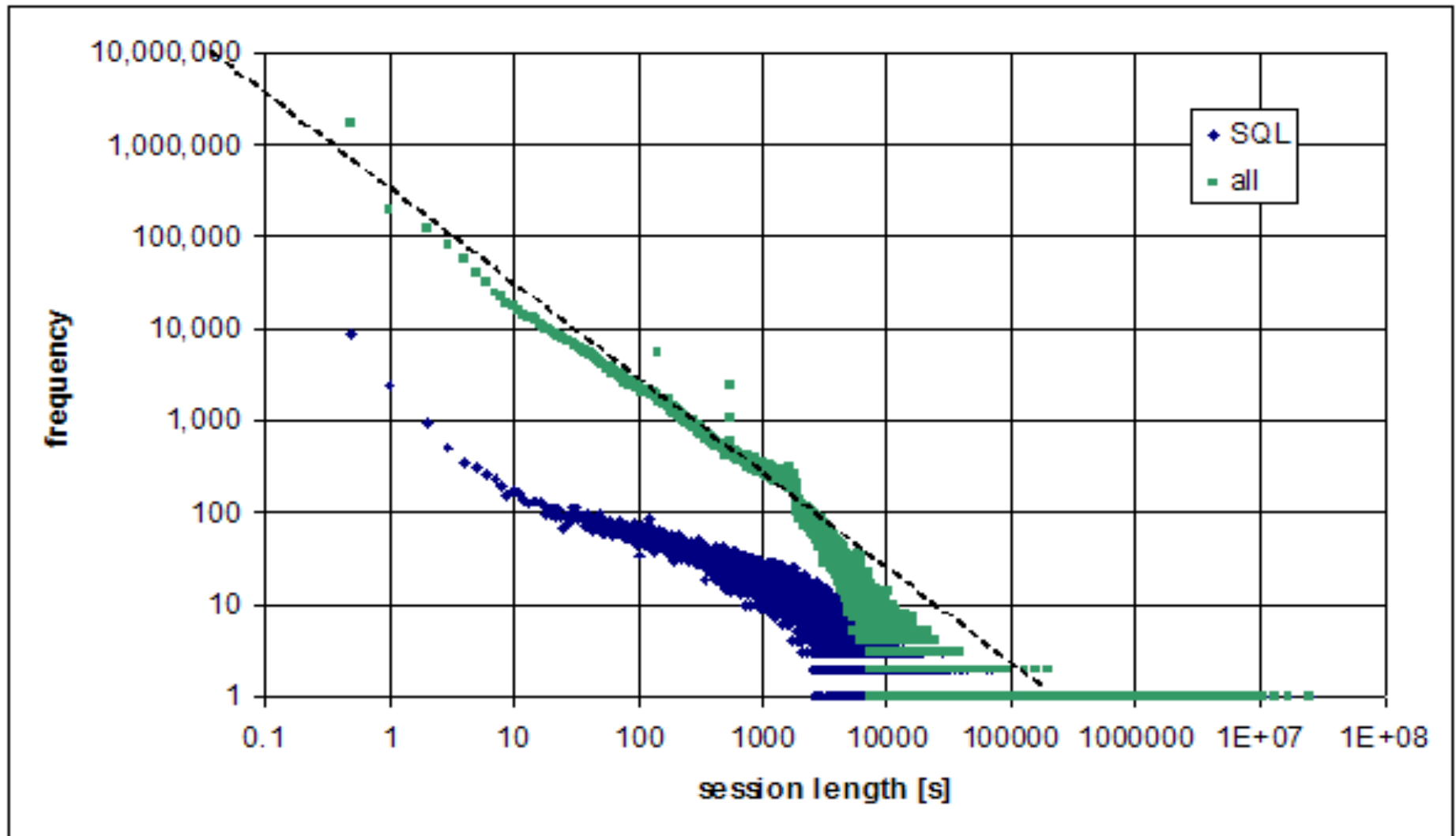
Support 1000s of users

E.g., Sloan Digital Sky Survey, ~10 TB; others much bigger





Skyserver Sessions (Thanks to Alex Szalay)





Who Will Host Your Services?

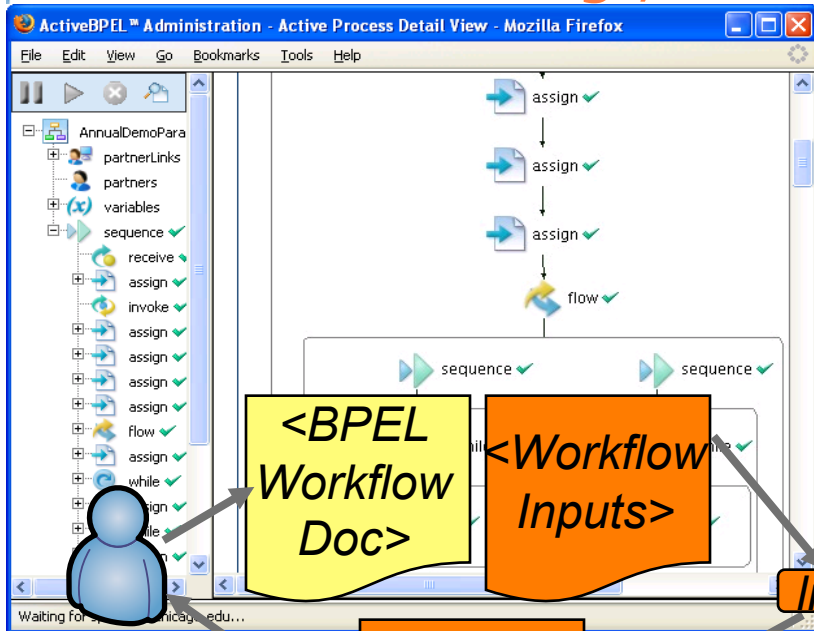
- Your institution (campus resources)
- (Inter)national systems
 - ◆ TeraGrid, Open Science Grid, UK Nat'l Grid Service, ChinaGrid, NaukaGrid, etc.
- Science domain systems
 - ◆ caBIG, NEES, Earth System Grid, Orion*, LEAD, NEON*, LHC Computing Grid, etc.
- Commercial systems
 - ◆ Amazon, Google, etc.



the globus alliance

www.globus.org

Composing Services: E.g., BPEL Workflow System



<BPEL Workflow Doc>

<Workflow Inputs>

<Workflow Results>

BPEL Engine

Data Service @ uchicago.edu

Analytic service @ duke.edu

Analytic service @ osu.edu

link

link

link

link

See also Kepler & Taverna



caBiG™ cancer Biomedical Informatics Grid™

an initiative of the National Cancer Institute

caBiG: <https://cabig.nci.nih.gov/>; BPEL work: Ravi Madduri et al.

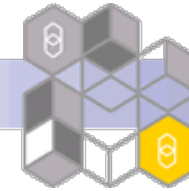


the globus alliance

www.globus.org

Composing Services

myGrid



Taverna Workbench v1.5.1.6

Design Results Discover

Search Watch loads

Local Services

- Notification Processor
- Local Java widgets
 - String Constant
 - BSF scripting host
 - AbstractProcessor - Processor for abstract taskdescriptions
 - RShell - Run R/S scripts through Rserve
 - Beanshell scripting host
- WSDL @ http://www.ebi.ac.uk/collab/mygrid/service1/goviz/GoViz.jws?wsdl
- WSDL @ http://eutils.ncbi.nlm.nih.gov/entrez/eutils/soap/eutils.wsdl
- WSDL @ http://soap.bind.ca/wsdl/bind.wsdl
- WSDL @ http://www.ebi.ac.uk/ws/services/urn:Dbfetch?wsdl
- WSDL @ http://soap.genome.jp/KEGG.wsdl
- WSDL @ http://www.ebi.ac.uk/xembl/XEMBL.wsdl
- Biomart service @ http://www.biomart.org/biomart
- Biomoby @ http://mobycentral.icapture.ubc.ca/cgi-bin/MOBY05/mobycentral.pl
- SeqHound @ seqhound.blueprint.org
- Soaplab @ http://www.ebi.ac.uk/soaplab/emboss4/services/

Advanced model explorer

Workflow Object properties

Workflow object	Retrie	Delay	Backof	Thread	Critica
BiomartAndEMBOSSAnalysis					
Workflow inputs					
Workflow outputs					
outputPlot					
HSapIDs					
MMusIDs					
RNorIDs					
Processors					
FlattenImageList	0	0	1	1	
getMMsequence	0	0	1	1	
getRNsequence	0	0	1	1	
getHSsequence	0	0	1	1	
hsapiens_gene_ensembl	0	0	1	1	
GetUniqueHomolog	0	0	1	1	
CreateFasta	0	0	1	1	
seqret	0	0	1	5	
emma	0	0	1	5	
plot	0	0	1	5	

Workflow diagram:

```

graph TD
    Input[hsapiens_gene_ensembl] --> GetUniqueHomolog[GetUniqueHomolog]
    GetUniqueHomolog --> getMMsequence[getMMsequence]
    GetUniqueHomolog --> getRNsequence[getRNsequence]
    GetUniqueHomolog --> getHSsequence[getHSsequence]
    getMMsequence --> CreateFasta[CreateFasta]
    getRNsequence --> CreateFasta
    getHSsequence --> CreateFasta
    CreateFasta --> seqret[seqret]
    seqret --> emma[emma]
    emma --> plot[plot]
    plot --> FlattenImageList[FlattenImageList]
    FlattenImageList --> outputPlot[outputPlot]
    GetUniqueHomolog --> HSapIDs[HSapIDs]
    GetUniqueHomolog --> MMusIDs[MMusIDs]
    GetUniqueHomolog --> RNorIDs[RNorIDs]
  
```

Workflow Outputs: outputPlot, HSapIDs, MMusIDs, RNorIDs

Rendering done.



OSGCC 2008

Globus Primer: An Introduction to Globus Software



Swift System

- Clean separation of logical/physical concerns
 - ◆ **XDTM** specification of logical data structures
- + Concise specification of parallel programs
 - ◆ **SwiftScript**, with iteration, etc.
- + Efficient execution on distributed resources
 - ◆ **Karajan** threading, **Falkon** provisioning, **Globus** interfaces, pipelining, load balancing
- + Rigorous provenance tracking and query
 - ◆ Virtual data schema & automated recording
- **Improved usability and productivity**
 - ◆ Demonstrated in numerous applications



Workflow Language - SwiftScript

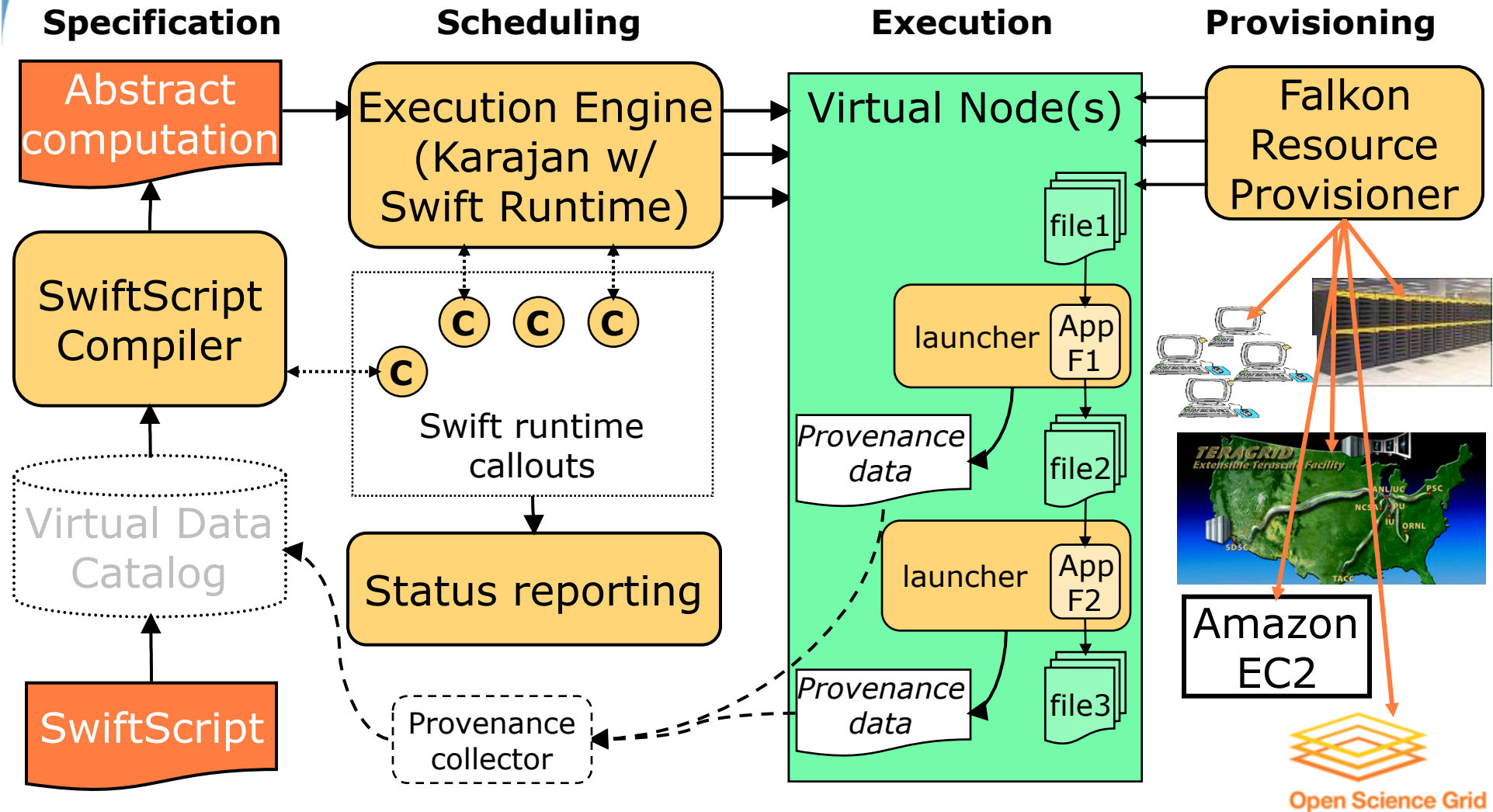
- Goal: Natural feel to expressing distributed applications
 - ◆ Variables (basic, data structures)
 - ◆ Conditional operators (if, foreach)
 - ◆ Functions (atomic / compound)
- Used to connect outputs to inputs
- It does not specify invocation order, only dependencies
- It can be seen as a metadata for expressing experiments



Execution Engine

- Karajan engine (event-based execution)
- Has a scheduler to map tasks to resources
 - ◆ Score-based planning
 - ◆ Recovers from failures (retries)
- Falkon resource manager creates a “virtual private cluster”
 - ◆ Uses Globus GRAM4 (PBS/Condor/Fork) to acquire resources from Grid systems

Dynamic Provisioning: Swift Architecture



Yong Zhao, Mihael Hatigan, Ioan Raicu, Mike Wilde, Ben Clifford

Using Globus to Locate Services

Case Study 1:

A Distributed Information Service for TeraGrid

John-Paul Navarro, Lee Liming



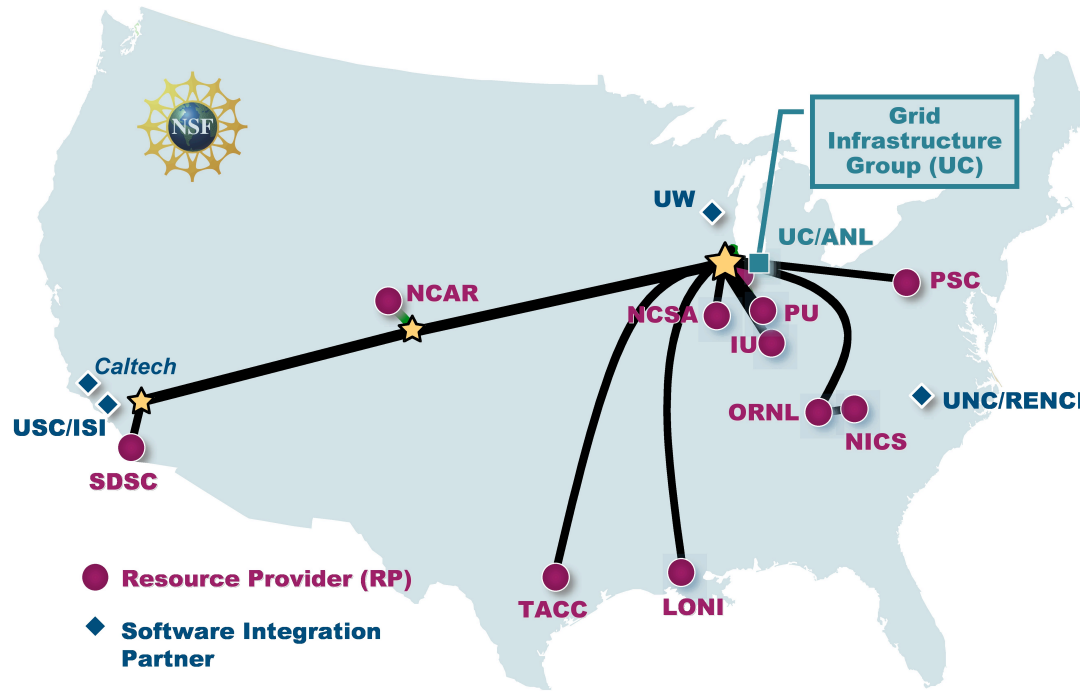
THE UNIVERSITY OF
CHICAGO



the globus alliance

www.globus.org

NSF's TeraGrid*



- **TeraGrid DEEP: Integrating NSF's most powerful computers (60+ TF)**

- ◆ 2+ PB Online Data Storage
- ◆ National data visualization facilities
- ◆ World's most powerful network (national footprint)

- **TeraGrid WIDE Science Gateways: Engaging Scientific Communities**

- ◆ 90+ Community Data Collections
- ◆ Growing set of community partnerships spanning the science community.
- ◆ Leveraging NSF ITR, NIH, DOE and other science community projects.
- ◆ Engaging peer Grid projects such as Open Science Grid in the U.S. as peer Grids in Europe and Asia-Pacific.

- **Base TeraGrid Cyberinfrastructure: Persistent, Reliable, National**

- ◆ Coordinated distributed computing and information environment
- ◆ Coherent User Outreach, Training, and Support
- ◆ Common, open infrastructure services

A National Science Foundation Investment in Cyberinfrastructure

\$100M 3-year construction (2001-2004)

\$150M 5-year operation & enhancement (2005-2009)

* Slide courtesy of Ray Bair, Argonne National Laboratory



The Challenge

- Provide a mechanism that allows resource providers, users, and partners the ability to publish and discover information about available capabilities
 - ◆ What are the TG compute resources?
 - ◆ What capabilities does resource X provide?
 - ◆ Where are the login services?
 - ◆ Where can I get data collection Y?
 - ◆ Who has a queue prediction service?
 - ◆ Who has a weather forecasting service?
- Provide a mechanism that is suitable for TeraGrid's open community
 - ◆ Publishers *register* information (as opposed to turning it over to a central database)
 - ◆ Central index (like Google) enables aggregation, discovery
 - ◆ Multiple access interfaces (WS/SOAP, WS/ReST, browser)



TG Information Services...

...IS NOT...	...IS...
A central database (Data Warehouse)	A central index/aggregation (Google)
A new user interface	A way user interfaces access information
A single implementation/tool	Includes several tools
A single software interface	Accessed using several useful interfaces
A specific set of data	Phased growing set of data
Changed data ownership	Ownership maintained as appropriate
Way to manage scientific information	Way to manage Grid meta-data
A data management system (database)	An information publishing system

...is a coordinated way to publish, index, and access public [Tera]Grid information using software interfaces.



Issues - Technical

- Information is stored in many legacy systems
 - ◆ Databases (several types, restricted access)
 - ◆ Static & dynamic web browser interfaces
- Schema are many and diverse
 - ◆ Impractical to design a relational database that supports all of these data types and relations
- Many kinds of clients (browsers, SOAP, ReST)
- High availability is critical
 - ◆ The service will be depended on both by TG operations (testing, documentation, planning) and by many TG users and partners, so it must be available all the time and very stable
 - ◆ Goal is 99.5% availability



Issues - Social

- TG is a community of independent service providers
 - ◆ Independence is prized
 - ◆ Ownership of information (and its quality control) is important
 - ◆ Participation in other grids is typical
- Publishers have low threshold for tech hassles
 - ◆ Publishing mechanism must be simple
- The solution must add to (not replace) existing interfaces

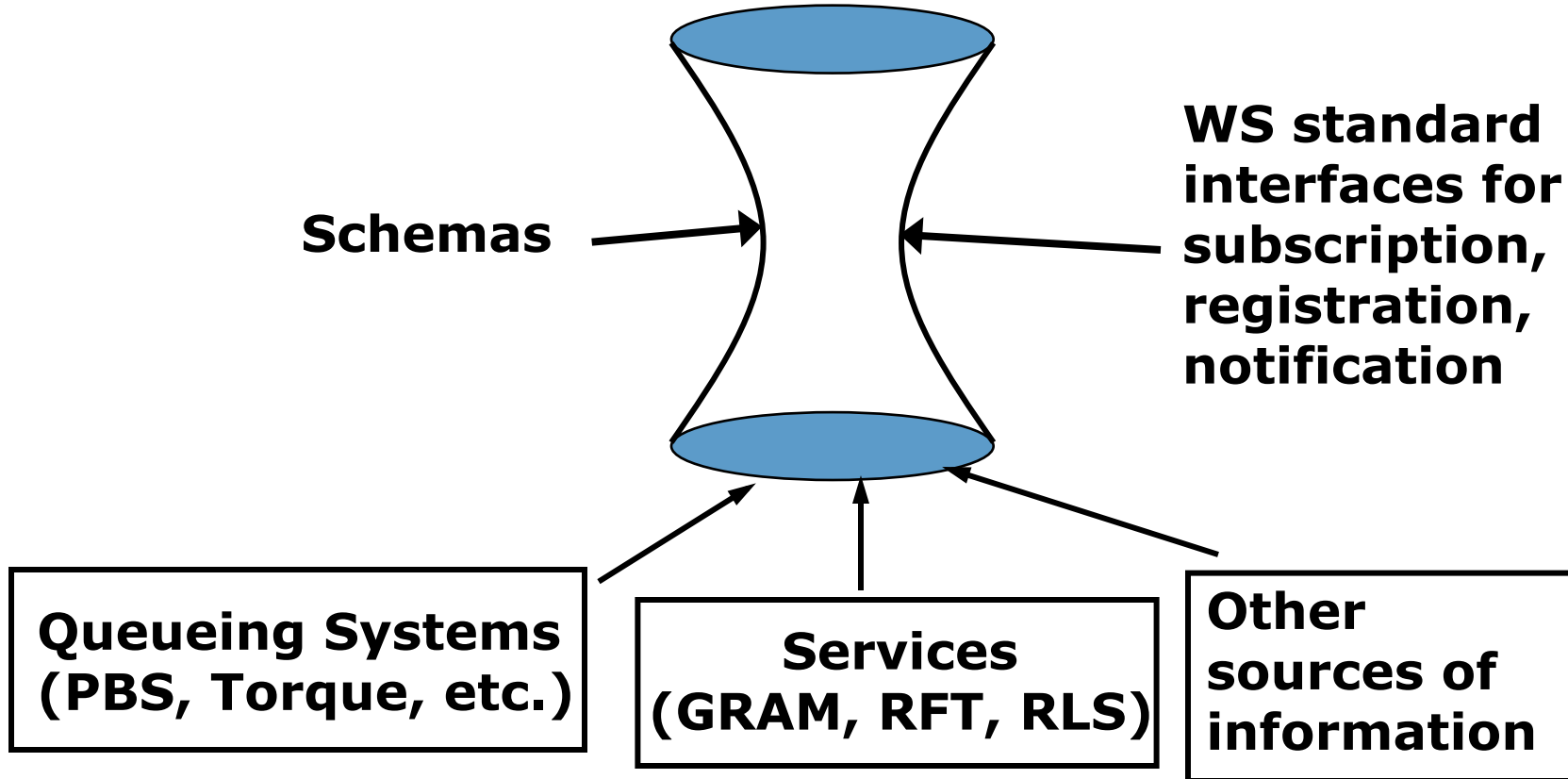


MDS4 Overview

- Components
 - ◆ Index Service – aggregates information and provides a query interface
 - ◆ Trigger Service – aggregates information and takes actions when conditions are met
 - ◆ WebMDS - subsets and transforms XML based on XPath queries, XSLT transforms and style sheets
 - ◆ Information provider APIs – integration with legacy systems
 - ◆ APIs and command-line clients for developers
- Implemented as Web services
- Uses WSRF (lifecycle, resource properties, etc.)
- Included in the Globus Toolkit 4.0

The MDS4 Hourglass

**Information Users :
Schedulers, Portals, Warning Systems, etc.**



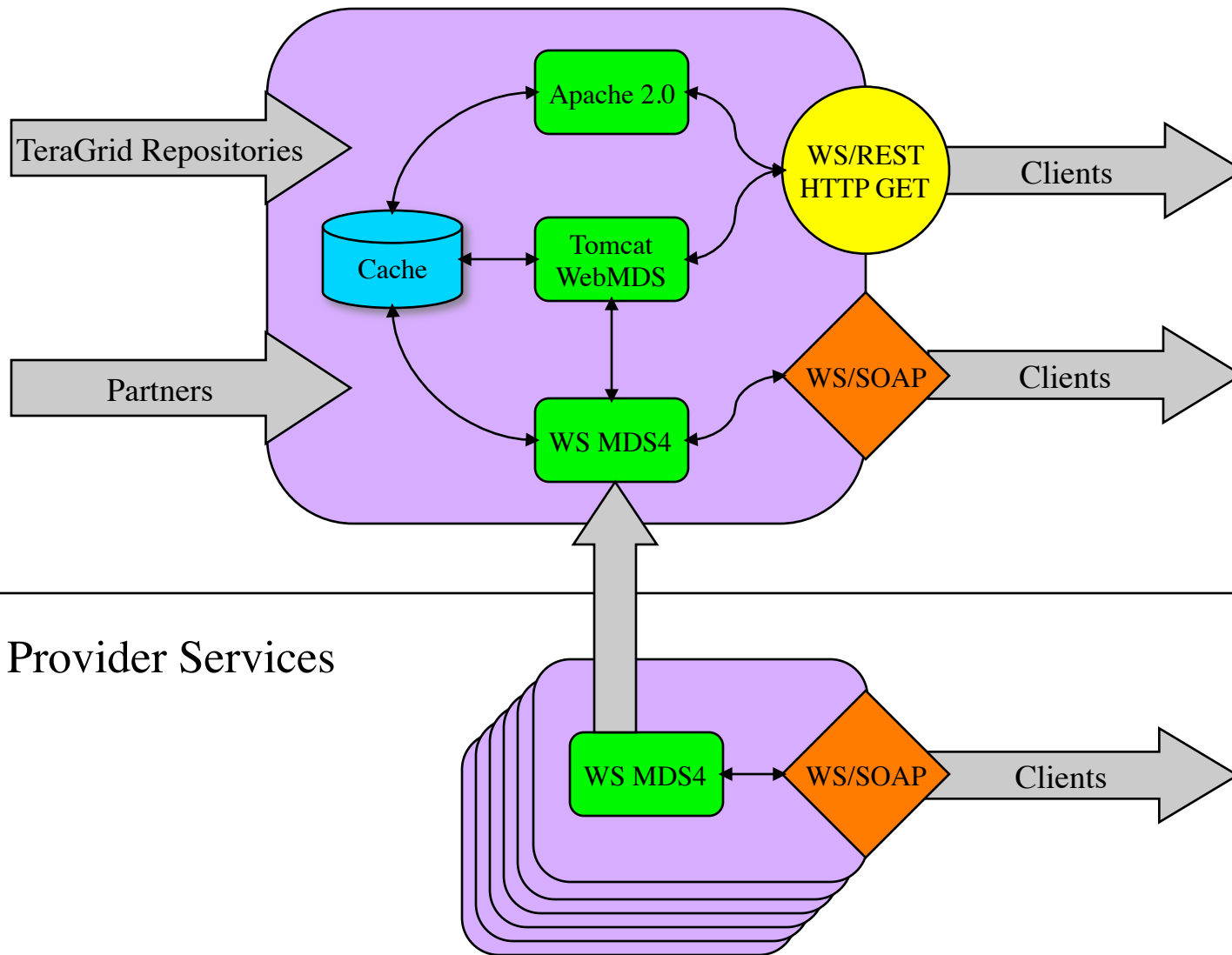


the globus alliance

www.globus.org

TeraGrid's IS Architecture

TeraGrid Central Services





Central vs. Distributed Services

- **Publisher Content**
 - ◆ Publisher-owned and maintained information
 - ◆ Data probably originates somewhere in the local system
- **Publisher Code**
 - ◆ An MDS4 index service
 - ◆ *Or:* Any Web service that has WS-ResourceProperties
- **Central Content**
 - ◆ Aggregated publisher content
- **Central Code**
 - ◆ Redundant servers
 - ◆ Information caching (persistence)
 - ◆ MDS4 index services (WS/SOAP)
 - ◆ WebMDS/Tomcat, Apache 2.0, ... (WS/REST)
 - ◆ Content published in: HTML, XHTML/XML, XML, Atom, RSS, ...

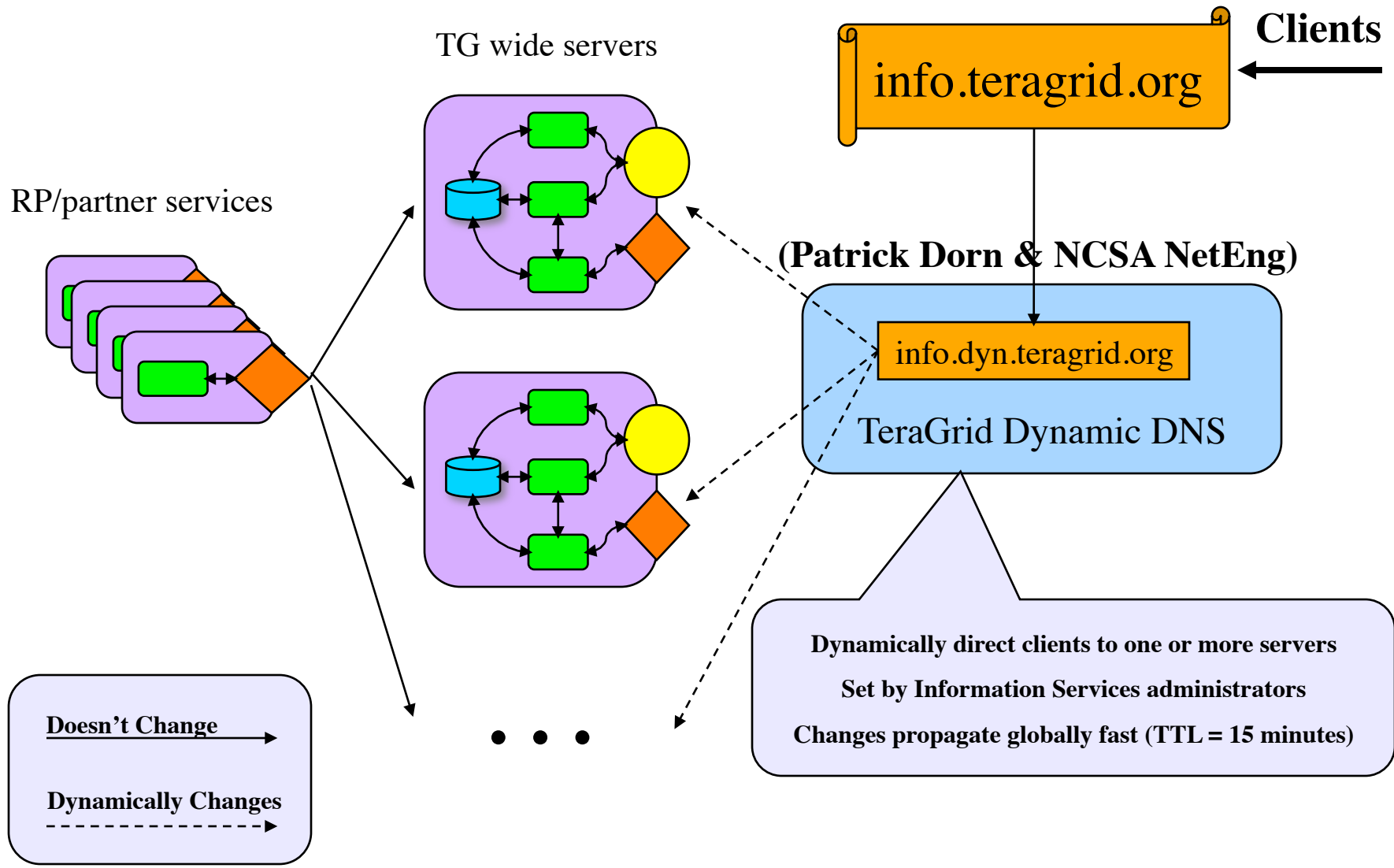


Registration

- Publisher *registers* available content
 - ◆ Local service maintains a *registration* with the central indices
 - ◆ Registration expires automatically, so refresh is needed periodically
 - ◆ Publishers retain ownership and operation of their own information service (can be registered with other grids!)
- Index services *pull* content
 - ◆ Registrations are subject to access control
 - ◆ Uses registration data to contact service and get latest content
 - ◆ Caches content locally, subject to purge policy
 - ◆ Cache allows for service outages, etc.



High-Availability Design



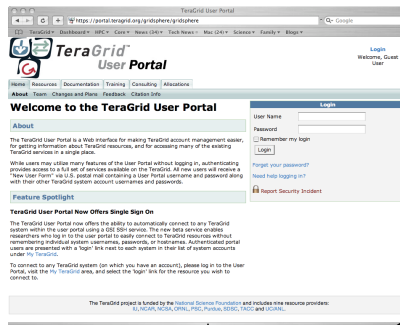


Information Services Users

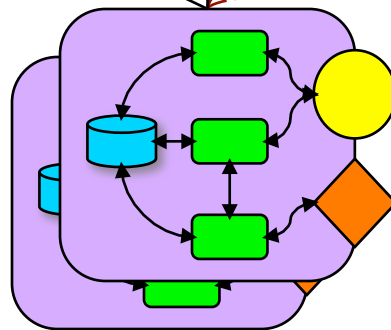
User Documentation



User Portal



Others



info.teragrid.org



Queue Contents in User Portal

TeraGrid User Portal

https://portal.teragrid.org:443/gridsphere/gridsphere?cid=systems-monitor&gs_act

TeraGrid Dashboard HPC Core News (33) Tech News Mac (17) Science Family Blogs

TeraGrid™ User Portal

Logout
Welcome, John-Paul Navarro

Home My TeraGrid Resources Documentation Training Consulting Allocations

Systems Monitor Science Gateways Data Collections HPC Queue Prediction [Beta] Remote Visualization [Beta] User Responsibilities

TeraGrid Systems Monitor

Back | Refresh

Job summary for login-abe.ncsa.teragrid.org:
[47 Running Jobs](#)
[68 Queued Jobs](#)
[50 Other Jobs](#)

Status	Job Id	Name	Owner	Queue	Submission Time	Processors
Running	35721.abem5.ncsa.uiu	d_1-40	petefred	normal		8
Running	35730.abem5.ncsa.uiu	l_1-39	petefred	normal		8
Running	36071.abem5.ncsa.uiu	rturb	pakshing	normal		16
Running	36518.abem5.ncsa.uiu	A2Q_8-8	seabra	normal		4
Running	36521.abem5.ncsa.uiu	A2Q_8-12	seabra	normal		4
Running	36523.abem5.ncsa.uiu	A2Q_12-12	seabra	normal		4
Running	36563.abem5.ncsa.uiu	AirNoFSTur	auzun	normal		30
Running	36584.abem5.ncsa.uiu	DI14	jhsin	normal		128
Running	36627.abem5.ncsa.uiu	rturb	pakshing	normal		16
Running	36647.abem5.ncsa.uiu	P3	amyshih	normal		24
Running	36690.abem5.ncsa.uiu	rturb	pakshing	normal		16
Running	36718.abem5.ncsa.uiu	cr-323-02	kjjin	normal		4
Running	36828.abem5.ncsa.uiu	m0r8_RC0s	moo	normal		16
Running	36833.abem5.ncsa.uiu	Estr_Prod	seabra	normal		16
Running	36842.abem5.ncsa.uiu	Script.abe	dcollins	normal		16
Running	36900.abem5.ncsa.uiu	L10_N64_4l	gbryan	normal		16
Running	36903.abem5.ncsa.uiu	frthall	zetienne	normal		150



the globus alliance

www.globus.org

Where are GridFTP services?

TeraGrid gridftp Services

http://info.teragrid.org/restdemo/html/tg/services/gridftp

<u>Version</u>	<u>Name</u>	<u>SiteID</u>	<u>ResourceID</u>	<u>Endpoint</u>	<u>Support Level->Goal</u>
4.0.5	gridftp-default-server	iu.teragrid.org	bigred.iu.teragrid.org	gsift://gridftp.bigred.iu.teragrid.org:2811/	production->production
4.0.5	gridftp-default-server	loni-lsu.teragrid.org	queenbee.loni-lsu.teragrid.org	gsift://qb1.loni.org:2811/	production->production
4.0.5	gridftp-default-server	ncar.teragrid.org	frost.ncar.teragrid.org	gsift://gridftp.frost.ncar.teragrid.org:2811/	production->production
4.0.5	gridftp-default-server	ncsa.teragrid.org	abe.ncsa.teragrid.org	gsift://gridftp-abe.ncsa.teragrid.org:2811/	production->production
4.0.5	gridftp-default-server	ncsa.teragrid.org	cobalt.ncsa.teragrid.org	gsift://gridftp-co.ncsa.teragrid.org:2811/	production->production
4.0.5	gridftp-default-server	ncsa.teragrid.org	dtf.ncsa.teragrid.org	gsift://gridftp-hg.ncsa.teragrid.org:2811/	production->production
4.0.5	gridftp-default-server	ncsa.teragrid.org	tungsten.ncsa.teragrid.org	gsift://gridftp-w.ncsa.teragrid.org:2811/	production->production
4.0.5	gridftp-default-server	ornl.teragrid.org	nstg.ornl.teragrid.org	gsift://tg-gridftp.ornl.teragrid.org:2811/	production->production
4.0.5	gridftp-default-server	psc.teragrid.org	bigben.psc.teragrid.org	gsift://gridftp.bigben.psc.teragrid.org:2811/	production->production
4.0.5	gridftp-default-server	purdue.teragrid.org	condor.purdue.teragrid.org	gsift://tg-data.purdue.teragrid.org:2811/	development->production
4.0.5	gridftp-default-server	purdue.teragrid.org	lear.purdue.teragrid.org	gsift://tg-data.purdue.teragrid.org:2811/	production->production
4.0.5	gridftp-default-server	purdue.teragrid.org	steele.purdue.teragrid.org	gsift://tg-data.purdue.teragrid.org:2811/	development->production
4.0.5	gridftp-default-server	sdsc.teragrid.org	datastar.sdsc.teragrid.org	gsift://ds-gridftp.sdsc.edu:2811/	production->production
4.0.5	gridftp-default-server	sdsc.teragrid.org	dtf.sdsc.teragrid.org	gsift://tg-gridftp.sdsc.teragrid.org:2811/	production->production
4.0.5	gridftp-default-server	sdsc.teragrid.org	intimidata.sdsc.teragrid.org	gsift://bg-login1.sdsc.edu:2811/	production->production
4.0.5	gridftp-default-server	uc.teragrid.org	dtf-rhel4.uc.teragrid.org	gsift://tg-gridftp.uc.teragrid.org:2811/	testing->testing
4.0.5	gridftp-default-server	uc.teragrid.org	dtf.uc.teragrid.org	gsift://gridftp.uc.teragrid.org:2811/	production->production
4.0.5	gridftp-default-server	uc.teragrid.org	viz.uc.teragrid.org	gsift://gridftp.uc.teragrid.org:2811/	production->production
4.0.5	gridftp-nonstriped-server	iu.teragrid.org	bigred.iu.teragrid.org	gsift://gridftp.bigred.iu.teragrid.org:2812/	production->production
4.0.5	gridftp-nonstriped-server	loni-lsu.teragrid.org	queenbee.loni-lsu.teragrid.org	gsift://qb1.loni.org:2811/	production->production



Where Can I Login?



Kit: login.teragrid.org version: 4.0.0

To login to cobalt at NCSA, ssh to grid-co.ncsa.teragrid.org!

Version	Name	SiteID	ResourceID	Endpoint	Support Level->Goal
3.9p1	gsi-openssh	ornl.teragrid.org	nstg.ornl.teragrid.org	tg-login.ornl.teragrid.org:22	production->production
4.2p1	gsi-openssh	iu.teragrid.org	bigred.iu.teragrid.org	login.bigred.iu.teragrid.org:22	production->production
4.5	gsi-openssh	ncar.teragrid.org	frost.ncar.teragrid.org	tg-login.frost.ncar.teragrid.org:22	production->production
4.5	gsi-openssh	ncsa.teragrid.org	abe.ncsa.teragrid.org	login-abe.ncsa.teragrid.org:22	production->production
4.5	gsi-openssh	ncsa.teragrid.org	cobalt.ncsa.teragrid.org	grid-co.ncsa.teragrid.org:22	production->production
4.5	gsi-openssh	ncsa.teragrid.org	dtf.ncsa.teragrid.org	login-hg.ncsa.teragrid.org:22	production->production
4.5	gsi-openssh	ncsa.teragrid.org	tungsten.ncsa.teragrid.org	login-w.ncsa.teragrid.org:22	production->production
4.5	gsi-openssh	psc.teragrid.org	bigben.psc.teragrid.org	tg-login.bigben.psc.teragrid.org:22	production->production
4.5	gsi-openssh	purdue.teragrid.org	condor.purdue.teragrid.org	tg-login.purdue.teragrid.org:22	production->production
4.5	gsi-openssh	purdue.teragrid.org	lear.purdue.teragrid.org	tg-login.purdue.teragrid.org:22	production->production
4.5	gsi-openssh	purdue.teragrid.org	steele.purdue.teragrid.org	tg-steele.purdue.teragrid.org:22	production->production
4.5	gsi-openssh	sdsc.teragrid.org	datastar.sdsc.teragrid.org	dslogin.sdsc.edu:22	production->production
4.5	gsi-openssh	sdsc.teragrid.org	dtf.sdsc.teragrid.org	tg-login1.sdsc.teragrid.org:22	production->production
4.5	gsi-openssh	sdsc.teragrid.org	intimidata.sdsc.teragrid.org	bg-login1.sdsc.teragrid.org:22	production->production
4.5	gsi-openssh	uc.teragrid.org	dtf-rhel4.uc.teragrid.org	tg-login.uc.teragrid.org:22	development->production
4.5	gsi-openssh	uc.teragrid.org	dtf.uc.teragrid.org	tg-login.uc.teragrid.org:22	production->production
4.5	gsi-openssh	uc.teragrid.org	dtf.uc.teragrid.org	tg-viz-login.uc.teragrid.org:22	production->production
4.5	gsi-openssh	uc.teragrid.org	viz.uc.teragrid.org	tg-viz-login.uc.teragrid.org:22	production->production
4.5p1	gsi-openssh	tacc.teragrid.org	lonestar.tacc.teragrid.org	tg-login.tacc.teragrid.org:22	production->production
4.5p1	gsi-openssh	tacc.teragrid.org	maverick.tacc.teragrid.org	tg-viz-login.tacc.teragrid.org:22	production->production
4.6	gsi-openssh	psc.teragrid.org	rachel.psc.teragrid.org	tg-login.rachel.psc.teragrid.org:22	production->production
4.7p1	gsi-openssh	tacc.teragrid.org	ranger.tacc.teragrid.org	ranger.tacc.teragrid.org:22	production->production

Don't use this one (yet).



Results - TeraGrid

- Considerable excitement from information owners...
 - ◆ A way to raise awareness for their information & capabilities
 - ◆ Doesn't require them to replace legacy systems or turn information over to someone else
- ...and information consumers
 - ◆ Simple, consistent access mechanisms for lots of information types
 - ◆ A mediating agency for independent service operators
- Integration to date:
 - ◆ Compute service descriptions and queue status
 - ◆ Software & service availability registry
 - ◆ Central documentation
 - ◆ Verification & validation testing service



MDS4 has...

- WS/WSRF interface
- WS/REST interface (browser-accessible)
- XSLT/Xpath support
- Registration, polling, subscription, notification capabilities
- Index & trigger service
- GLUE CE providers
- Plug-in API for custom info providers

MDS4 doesn't have...

- Your own custom info providers
- Schema validation
- Many clients (unless you count browsers!)
- XSLT style examples
- High-availability deployment



Other Uses of MDS4

- Directory of service deployments
 - ◆ E.g., caGrid service registry
- Monitoring/alert service
 - ◆ Trigger service notifies when an expected service registration isn't there anymore
- Monitoring/recording service
 - ◆ Subscriber periodically records value of a registered resource property (e.g., free space, services registered, system load)

Using Globus to Share Data

Case Study 3:

Data Replication for LIGO

Scott Koranda, Ann Chervenak



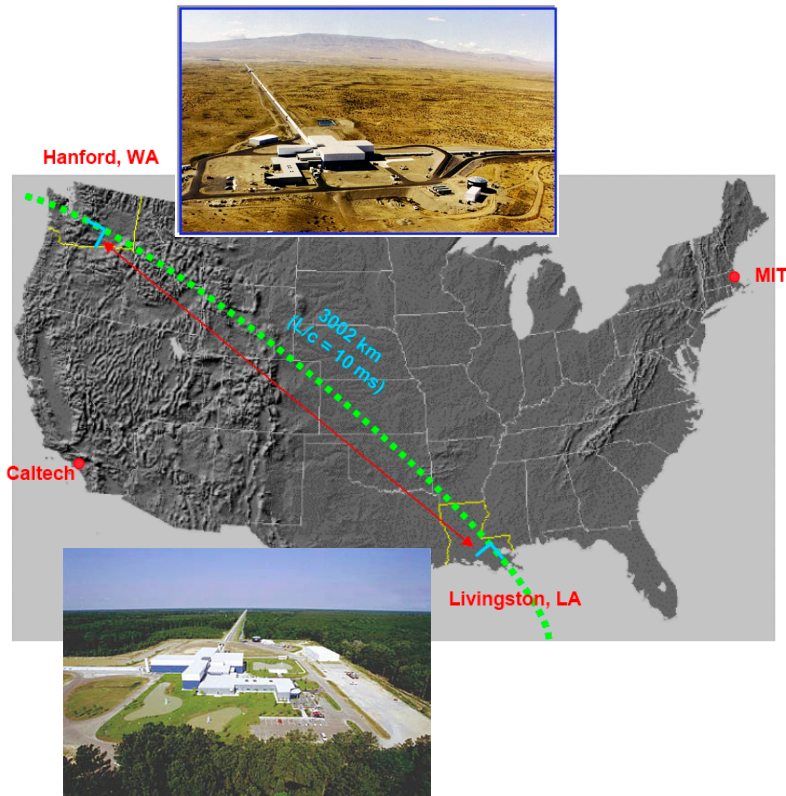
THE UNIVERSITY OF
CHICAGO



the globus alliance

www.globus.org

Laser Interferometer Gravitational Wave Observatory



- Goal: Observe gravitational waves predicted by theory
- Three physical detectors in two locations (plus GEO detector in Germany)
- 10+ data centers for data analysis
- Collaborators in ~ 40 institutions on at least three continents



LIGO (in 2005) by the Numbers

- In 2005, LIGO was recording thousands of channels, generating approx. 1 TB per day of data during a detection run
 - ◆ Data is published and data centers subscribe to portions that local users want for analysis or for local storage
- Data analysis results in derived data
 - ◆ In 2005, this was ~30% of all LIGO data
 - ◆ This also is published and replicated
- Over 30 million files in LDR network (April 2005)*
 - 6+ million unique logical files
 - 30+ million physical copies of those files

* Scott Koranda, UW-Milwaukee, April 2005

The Challenge

Replicate 1 TB/day of data to 10+ international sites

- ◆ Publish/subscribe (pull) model
- ◆ Provide scientists with the means to specify and discover data based on application criteria (metadata)
- ◆ Provide scientists with the means to locate copies of data



Issues - Technical

- Efficiency
 - ◆ Avoid unused bandwidth while data transfers are taking place, esp. on high-bandwidth links (10+ Gbps)
 - ◆ Avoid idle time on the network between transfers



Issues - Social (1)

- Workflow
 - ◆ The publish/subscribe model matches how scientists think about their use of the data
 - ◆ Use of metadata is critical
- Security
 - ◆ Authenticate endpoints of data transfers to prevent unauthorized data manipulation
- Heterogeneity
 - ◆ Can't tell data centers what storage systems to use
 - ◆ Can't get everyone to do accounts the same way

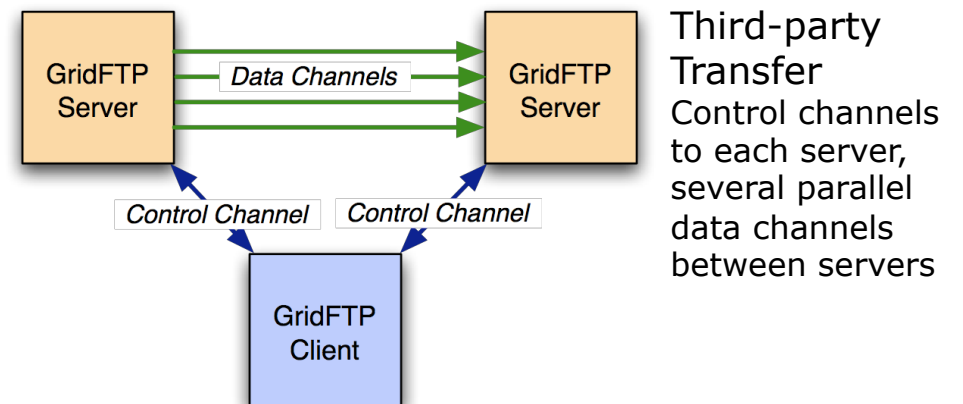
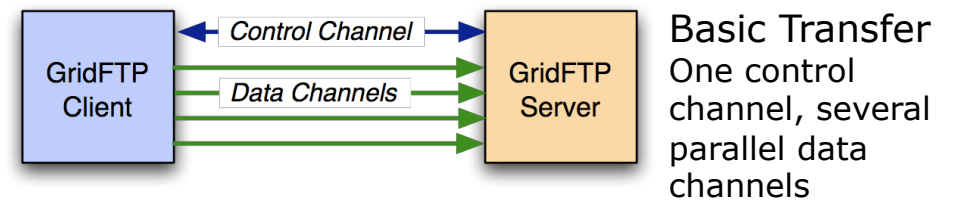


Issues - Social (2)

- Maintenance
 - ◆ LIGO is focused on science, not IT
 - ◆ Ideally, they would not have to build or maintain data management software
 - ◆ GOAL: If software must be produced, keep it simple and do it in a way that non-CS people can understand
 - ◆ GOAL: Produce a solution that others will adopt and maintain for them

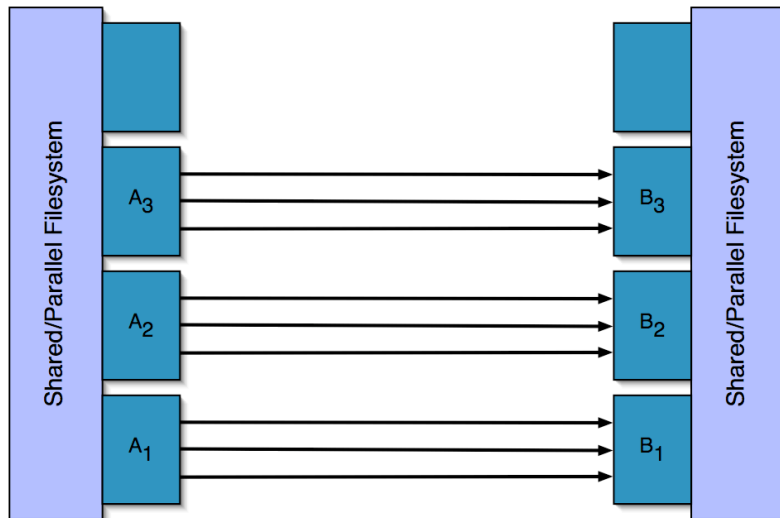
GridFTP

- A high-performance, secure data transfer service optimized for high-bandwidth wide-area networks
 - ◆ FTP with extensions
 - ◆ Uses basic Grid security (control and data channels)
 - ◆ Multiple data channels for parallel transfers
 - ◆ Partial file transfers
 - ◆ Third-party (direct server-to-server) transfers
- GGF recommendation GFD.20





Striped GridFTP



- GridFTP supports a striped (multi-node) configuration
 - ◆ Establish control channel with one node
 - ◆ Coordinate data channels on multiple nodes
 - ◆ Allows use of many NICs in a single transfer
- Requires shared/parallel filesystem on all nodes
 - ◆ On high-performance WANs, aggregate performance is limited by filesystem data rates



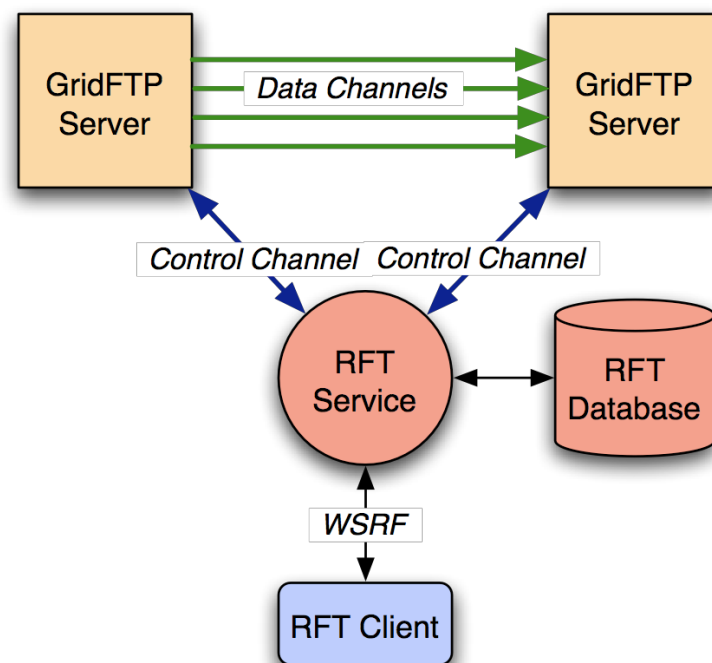
globus-url-copy

- Command-line client for GridFTP servers
 - ◆ Text interface
 - ◆ No “interactive shell” (single command per invocation)
- Many features
 - ◆ Grid security, including data channel(s)
 - ◆ HTTP, FTP, GridFTP
 - ◆ Server-to-server transfers
 - ◆ Subdirectory transfers and lists of transfers
 - ◆ Multiple parallel data channels
 - ◆ TCP tuning parameters
 - ◆ Retry parameters
 - ◆ Transfer status output



RFT - File Transfer Queuing

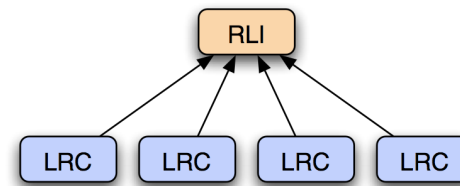
- A WSRF service for queuing file transfer requests
 - ◆ Server-to-server transfers
 - ◆ Checkpointing for restarts
 - ◆ Database back-end for failovers
- Allows clients to request transfers and then “disappear”
 - ◆ No need to manage the transfer
 - ◆ Status monitoring available if desired



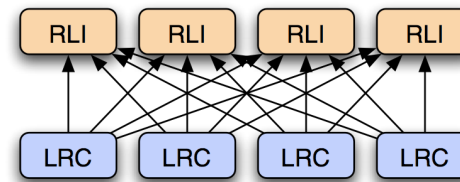


RLS - Replica Location Service

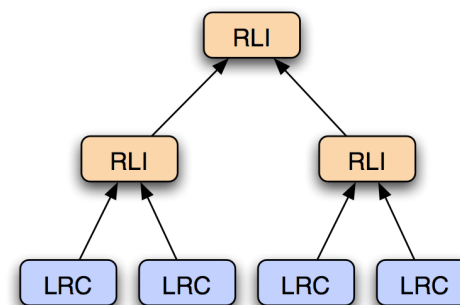
- A distributed system for tracking replicated data
 - ◆ Consistent local state maintained in Local Replica Catalogs (LRCs)
 - ◆ Collective state with relaxed consistency maintained in Replica Location Indices (RLIs)
- Performance features
 - ◆ Soft state maintenance of RLI state
 - ◆ Compression of state updates
 - ◆ Membership and partitioning information maintenance



Simple Hierarchy
The most basic deployment of RLS



Fully Connected
High availability of the data at all sites



Tiered Hierarchy
For very large systems and/or very large collections

pyGlobus*

- High-level, object-oriented interface in Python to GT4 Pre-WS APIs.
 - ◆ GSI security
 - ◆ GridFTP
 - ◆ GRAM
 - ◆ XIO
 - ◆ GASS
 - ◆ MyProxy
 - ◆ RLS
- Also includes tools and services
 - ◆ GridFTP server
 - ◆ GridFTP GUI client
 - ◆ Other GT4 clients

* pyGlobus contributed to GT4 by the Distributed Computing Department of LBNL



Lightweight Data Replicator*

Ties together 3 basic Grid services:

1. Metadata Service

- info about files such as size, md5, GPS time, ...
- sets of interesting metadata propagate
- answers question "What files or data are available?"

2. Globus Replica Location Service (RLS)

- catalog service maps filenames to URLs
- also maps filenames to sites
- answers question "Where are the files?"

3. GridFTP Service

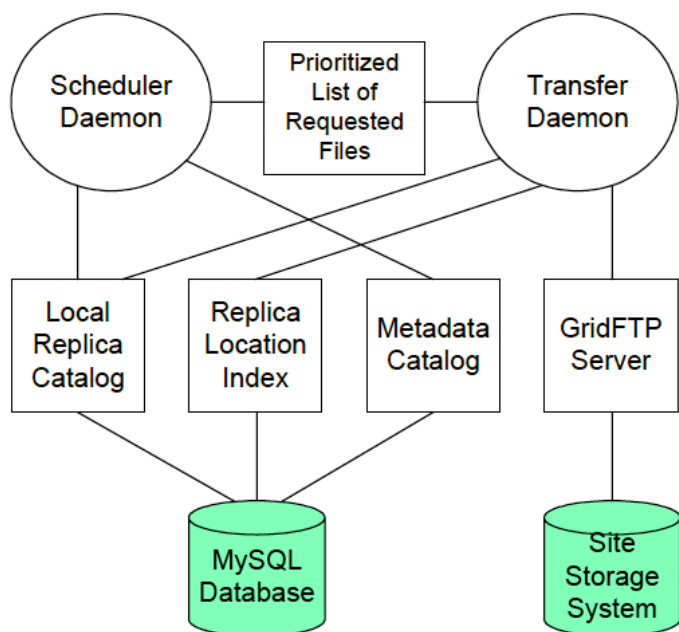
- server and customized, tightly integrated client
- use to actually replicate files from site to site
- answers question "How do we move the files?"



* Slide courtesy of Scott Koranda, UW-Milwaukee



LDR Architecture



- Each site has its own machinery for pulling data needed to satisfy local users
- Scheduler daemon queries metadata and replica catalogs to identify missing files, which it records in a prioritized list
- Transfer daemon manages transfers for files on the list, and updates LRC when files arrive
- If a transfer fails, the transfer daemon simply re-adds the file to the list for future transfers
- Daemons and metadata service written in Python



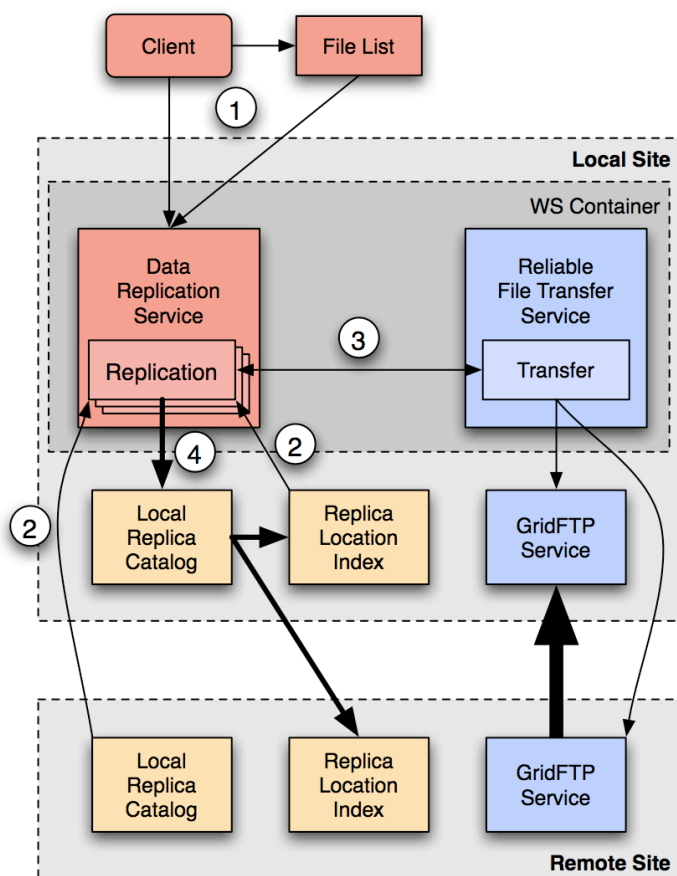
Results - LIGO*

- LIGO/GEO S4 science run completed March
 - ◆ Replicated over 30 TB in 30 days
 - ◆ Mean time between failure now one month (outside of CIT)
 - ◆ Over 30 million files in LDR network
 - 6+ million unique LFNs in RLS network
 - 30+ million PFNs for those LFNs
- Performance currently appears to be limited by Python performance
- Partnership with Globus Alliance members is leading to LDR-like tools being added to Globus Toolkit

* Slide courtesy of Scott Koranda, UW-Milwaukee



Epilogue: Data Replication Service



- Data Replication Service (DRS)
 - ◆ Reimplementation of LDR's subscription capabilities
 - ◆ Uses WSRF-compliant services written in Java
 - ◆ Works with RLS and RFT services in GT4
- Tech preview in GT4
- LIGO evaluating as a replacement for LDR's subscription system
- Remaining piece is metadata service



Globus has...

- High-performance data movement
- WS/WSRF interface
- Java client
- Command-line client
- Replica location
- Reliable replication
- Database integration (OGSA DAI)
- Plug-in interface for mass storage systems

Globus doesn't have...

- Metadata services
- Interactive text client (see UberFTP)
- GUI client (see SGGC incubator)
- File synchronization
- Automatic tuning
- Wide-area filesystem
- High-availability deployment



Other Uses

- GridFTP can be embedded in applications for high-performance data streaming
- GridFTP can be used with SSH-style public keys instead of certificates
- RFT can provide a Web services interface to GridFTP
- RFT is used by GRAM for file staging
- OGSA DAI can be used to implement a metadata service
- And many more...

Using Globus for Massive Data Analysis

Case Study 2:

High-Throughput Data Analysis for GEO600

Thomas Radke, Stuart Martin



THE UNIVERSITY OF
CHICAGO



the globus alliance

www.globus.org

GEO600 Observatory



- Goal: Same as LIGO: observe gravitational waves predicted by theory
- 600m laser interferometer near Hannover, Germany
- Members of the LIGO Scientific Collaboration



The Challenge

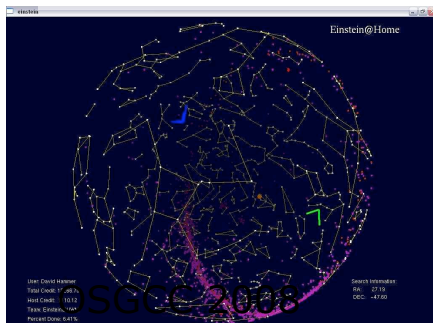
- Sift through the data produced by the GEO600 instrument to find evidence of a GW signal
 - ◆ Very complicated signal analysis task
 - ◆ An overwhelming amount of data to review
 - ◆ No single huge computer anywhere, but several clusters available at data centers
 - ◆ D-GRID and OSG resources available



Dual Approach

Einstein@Home

- Shared with LIGO community
- Runs on volunteer user desktop/laptop systems
- Uses BOINC network
- Running since mid-2006
- >70,000 computers participating/week
- ~19000 units/day



AstroGrid-D

- Same scientific application as Einstein@Home
- Uses D-Grid and OSG resources
- Running since Oct 07
- All jobs submitted using GRAM4 (globusrun-ws)
- Averaging ~4000 jobs per day

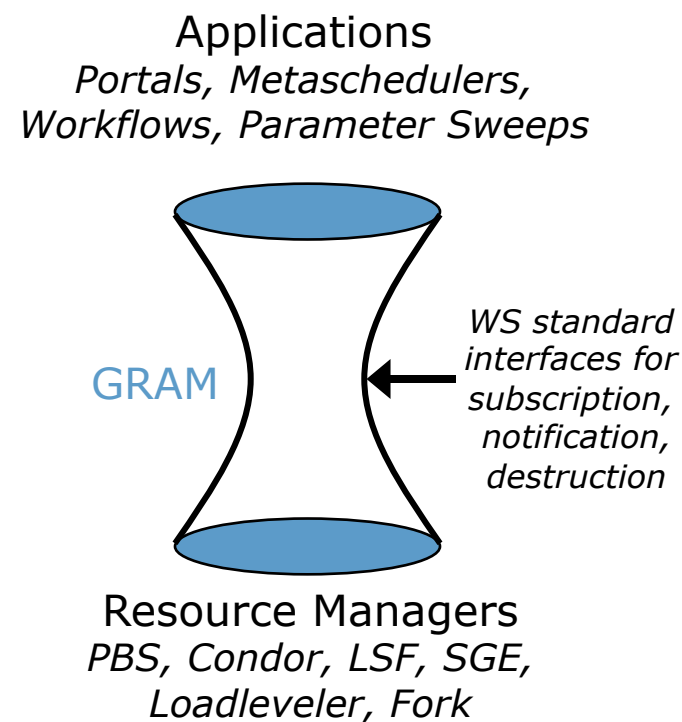


Traditional Resource Management Approach

- Have access to numerous sites
 - ◆ Accounts, permissions, etc
- Use a Metascheduler to make resource selection decisions
 - ◆ GridWay
 - ◆ Metascheduler uses GRAM to contact the difference local queuing systems

GRAM – Remote Job Submission and Control Service

- A remote job submission and control service
 - ◆ Includes file staging and I/O management
 - ◆ Includes reliability features
 - ◆ Supports basic Grid security mechanisms
 - ◆ Available in Pre-WS and WS
- GRAM is *not* a scheduler
 - ◆ No scheduling
 - ◆ No metascheduling/brokering
 - ◆ Often used as a front-end to schedulers, and often used to simplify metaschedulers/brokers





GRAM4 Scalability

- Scalability a major focus of GRAM's design
 - ◆ GRAM4 can manage 32,000 active jobs
 - ◆ Ability to manage load on control node
 - ◆ GRAM4 can handle bursts of up to 50 job submissions
 - ◆ Each job requires ~ 2 s to process
- Are the error conditions acceptable?
 - ◆ Job should be rejected or timeout before overloading the service container or service host



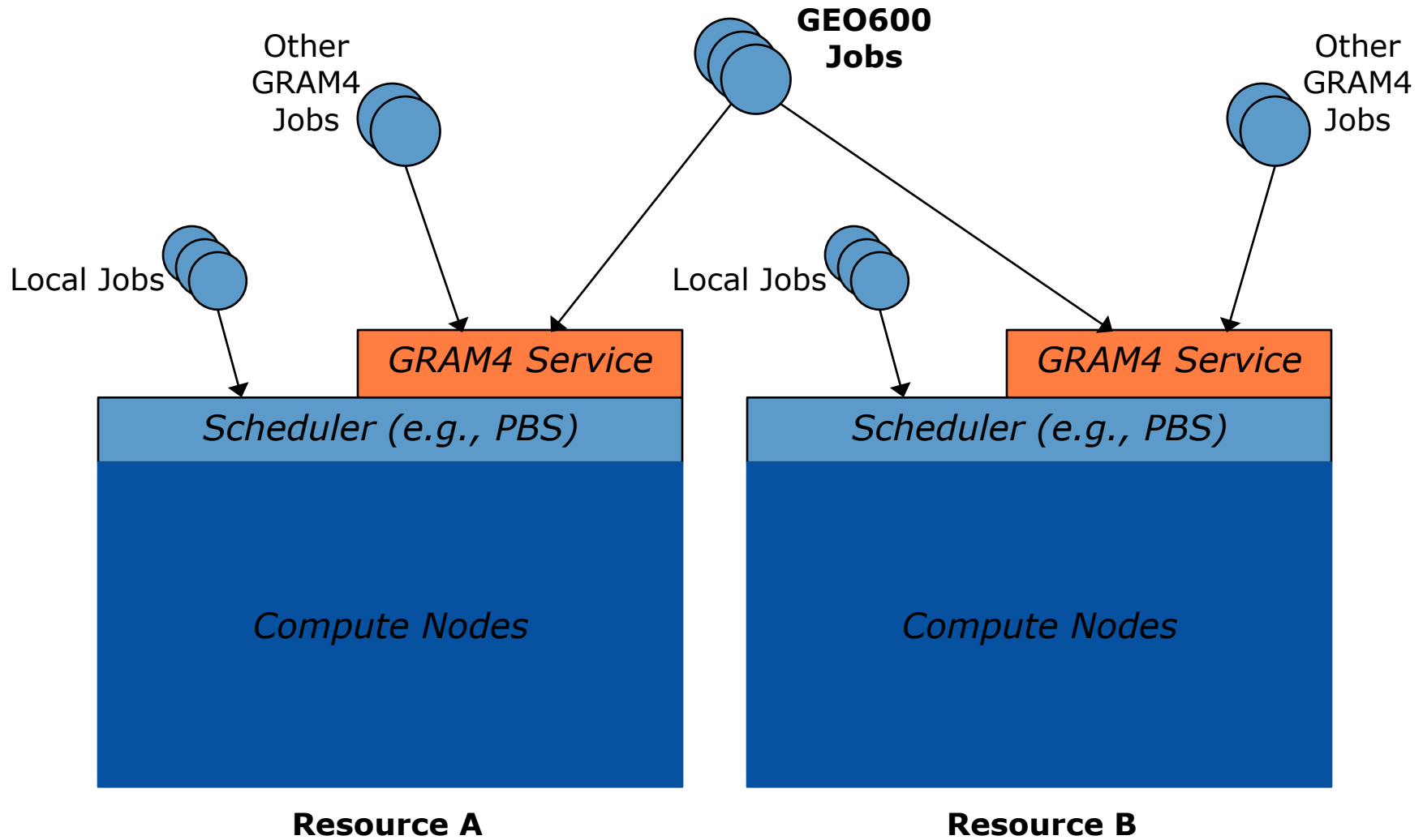
GEO600 Approach

- Submission host at AEI has list of all Grid resources we want to run GEO600 on with max/min jobs for each
- Hourly cron job reviews list of resources and jobs submitted, queries status of all submitted jobs
- For completed jobs, stdout/stderr and logfiles are staged back to the submission machine
- When resource has fewer than N jobs, more jobs (up to max) are submitted using globusrun-ws client, input files are staged in
- EPR of each job is kept on the submission machine for later job status queries

~4000 jobs/day processed this way!



D-Grid's Perspective






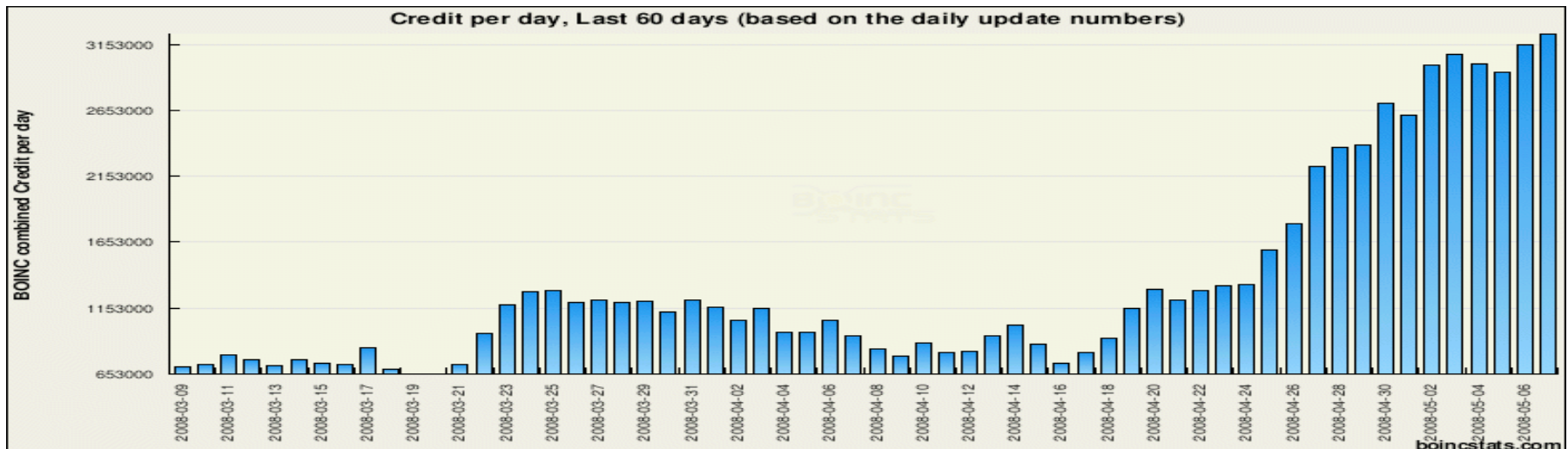
the globus alliance

www.globus.org

AstroGrid-D Performance

- #1 as reported on Einstein@home top users
 - ◆ http://einstein.phys.uwm.edu/top_users.php

Rank	Name	Recent average credit	Total credit	Country	Participant since
1	AEI eScience group, for the German Grid (D-Grid) and the Open Science Grid (OSG)	2,392,379.26	135,298,045	Germany	1 Feb 2007 17:05:25 UTC
2	Steffen Grunewald, for Merlin/Morgane 	766,795.98	206,122,300	Germany	18 Oct 2004 23:36:26 UTC





Globus has...

- Web service for job submission and control
- Cmd line and Java clients
- Data stage-in/stage-out
- Client notification support
- Plug-in interface for local resource managers
- Support for popular resource managers
- Plug-in interface for authorization decisions
- Advance reservation support (GARS)
- Standards compliance
- Supported on natl. grids

Globus doesn't have...

- Fancy submission tools (see Condor-G)
- Portlets (see CoG, OGCE)
- Scheduling/queuing (see GridWay, PBS, LSF...)
- Co-scheduling (see GARS, HARC, GUR)
- Support for every resource manager
- Support for complicated authorization decisions (see VOMS, CAS, GridShib)



Other Uses

- Simplify the development of science gateways (See: LEAD portal, GridChem, NCBioPortal...)
- Simplify the implementation of a metascheduler (e.g., GridWay)
- Support applications driven by workflow engines (e.g., Swift, Taverna, BPEL)
- Enable application/service hosting by submitting VM images as jobs
- Build a virtual cluster by submitting job proxies and registering allocated nodes (see MyCluster)

Using Globus to Scale an Application

Case Study 4:

Scientific Workflow for Computational Economics

Tiberiu Stef-Praun, Gabriel Madeira, Ian Foster,
Robert Townsend



THE UNIVERSITY OF
CHICAGO



The Challenge

- Expand capability of economists to develop and validate models of social interactions at large scales
 - ◆ Harness large computation systems
 - ◆ Simplify programming model (eye toward easy integration of science code)
 - ◆ Improve automation
- Requires an end-to-end approach, but through integration, not the “silo” model



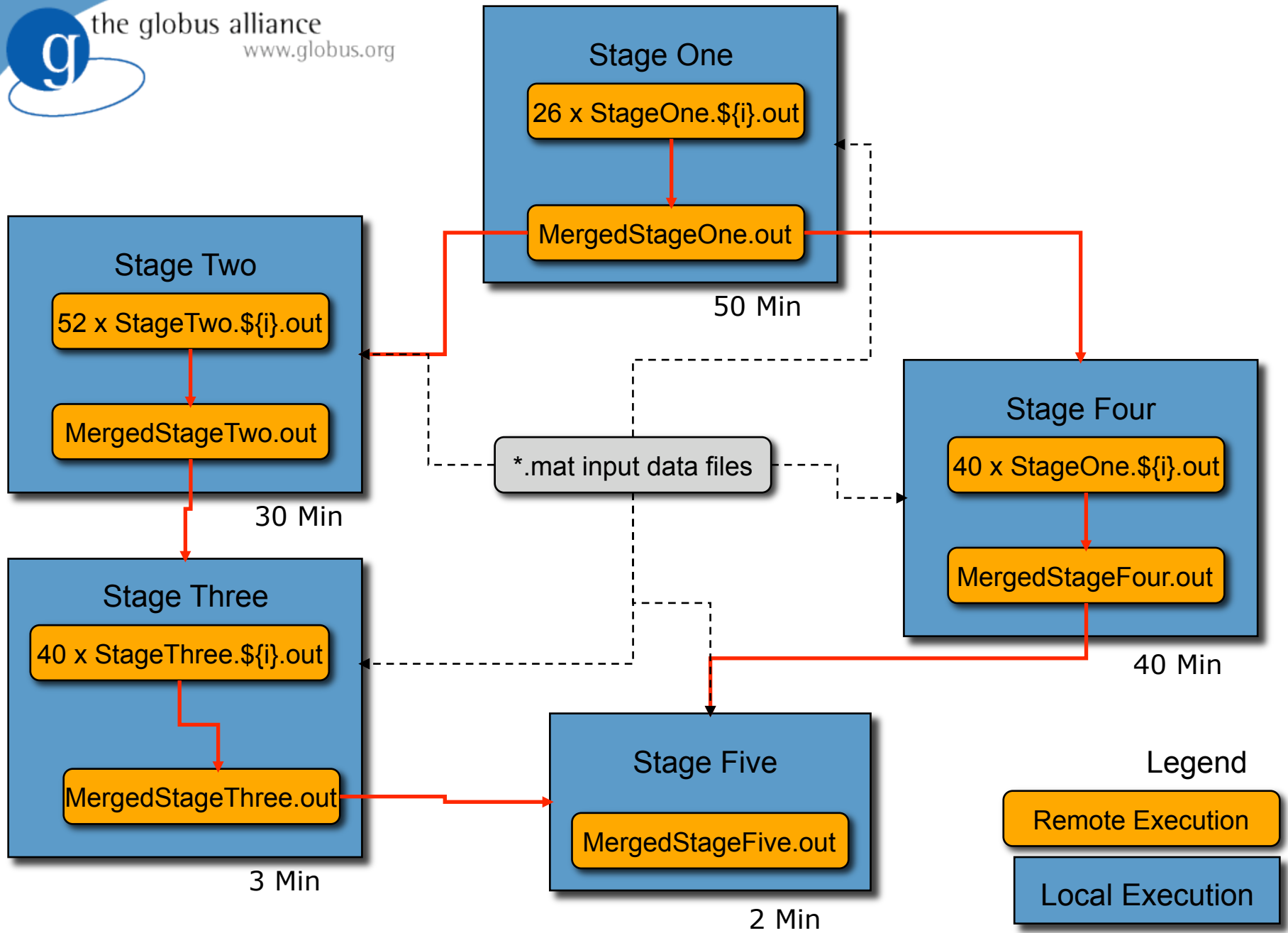
Moral Hazard Problem

- An entity in control of some resources (the entrepreneur) contracts with other entities that use these resources to produce outputs (the workers)
- Two organizational forms are available
 - ◆ The workers cooperate on their efforts and divide up their income (thus sharing risks)
 - ◆ The workers are independent of each other, and are rewarded based on relative performance
- Both are stylized versions of what is observed in tenancy data in villages such as in Maharashtra, India (Townsend and Mueller 1998)



Moral Hazard Solver

- Five stages, each solved by linear programming
 - ◆ Balance between promises for future and consumption to optimally reward agents
- In each stage: Given a set of parameters: consumption, effort, technology, output, wealth
 - ◆ Do a linear optimization to find out the best behavior
 - ◆ Parameter sweep (grid of parameter values)
 - ◆ Linear solver is run independently on each point of the parameter grid
 - ◆ Results are merged at end of the stage
- Across stages: Different organization (parameters) for similar stage structure
 - ◆ Most stages depend on results of other stages





Issues - Technical

- Language
 - ◆ Science code written in MATLAB/Octave
 - ◆ End to end system must be language-independent
- Code prerequisites
 - ◆ Each solver task requires MATLAB/Octave pre-installed on the execution node, and solver code staged in prior to execution
 - ◆ Each solver task requires files from previous stages
- Automation
 - ◆ ~200 tasks must be executed
 - ◆ This is a lot of “babysitting” if performed manually



Issues - Social

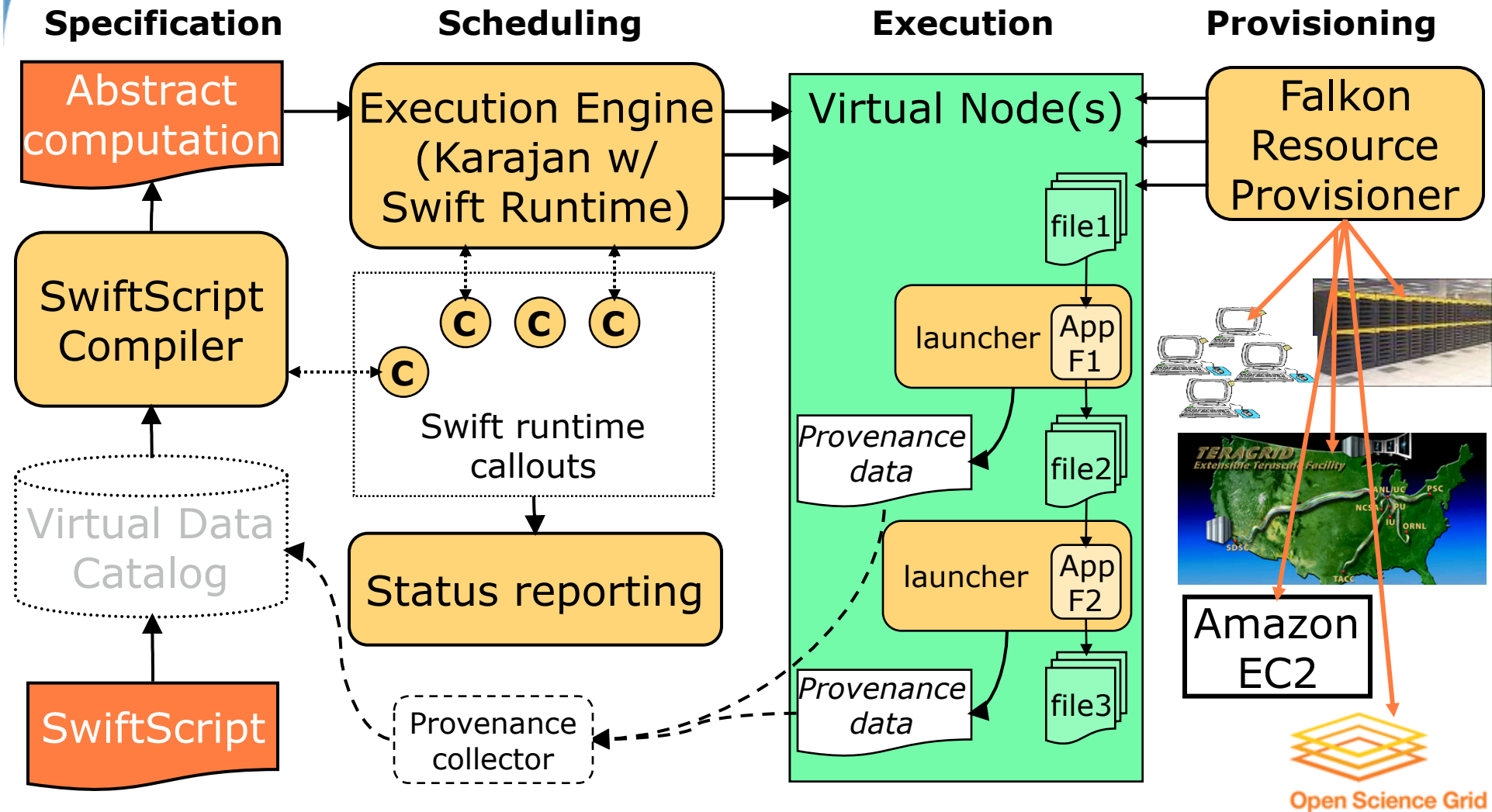
- **Licensing**
 - ◆ MATLAB licensing has a per-node cost
 - ◆ Expensive if you're using $O(10)+$ nodes
- **Provenance**
 - ◆ Task execution, data integrity
 - ◆ Not a huge concern at this scale, but for larger scales (10,000 tasks) it is important to record how the work is performed
- **Provisioning, resource sharing**
 - ◆ This problem used a shared campus cluster (at U Chicago)
 - ◆ We know of problems with 2-3 orders of magnitude more tasks, which require (inter)national-scale resources to accomplish in a timely fashion



Swift System

- Swift is a Grid-enabled application framework
 - ◆ Emphasis on workflow and adapting legacy application to a Grid environment
- Technical features
 - ◆ Clean separation of logical/physical concerns
 - **XDTM** specification of logical data structures
 - + Concise specification of parallel programs
 - **SwiftScript**, with iteration, etc.
 - + Efficient execution on distributed resources
 - **Karajan** threading, **Falkon** provisioning, **Globus** interfaces, pipelining, load balancing
 - + Rigorous provenance tracking and query
 - Virtual data schema & automated recording
 - **Improved usability and productivity**
 - Demonstrated in numerous applications

Dynamic Provisioning: Swift Architecture



Yong Zhao, Mihael Hatigan, Ioan Raicu, Mike Wilde, Ben Clifford



Workflow Language - SwiftScript

- Goal: Natural feel to expressing distributed applications
 - ◆ Variables (basic, data structures)
 - ◆ Conditional operators (if, foreach,)
 - ◆ Functions (atomic / compound)
- Used to connect outputs to inputs
- It does not specify invocation order, only dependencies
- It can be seen as a metadata for expressing experiments



Execution Engine

- Karajan engine (event-based execution)
- Has a scheduler to map tasks to resources
 - ◆ Score-based planning
 - ◆ Recovers from failures (retries)
- Falkon resource manager creates a “virtual private cluster”
 - ◆ Uses Globus GRAM4 (PBS/Condor/Fork) to acquire resources from Grid systems



The Solution

- Code changes
 - ◆ Solver code was broken into modules (atomic blocks) to allow parallel execution
 - ◆ Code ported from MATLAB to Octave to avoid per-node licensing fees
 - ◆ Workflow was described in SwiftScript
- Software installation
 - ◆ Swift engine, Karajan, Falkon deployed locally
- Shared resource (already available)
 - ◆ Existing compute cluster with GRAM4, GridFTP, etc.



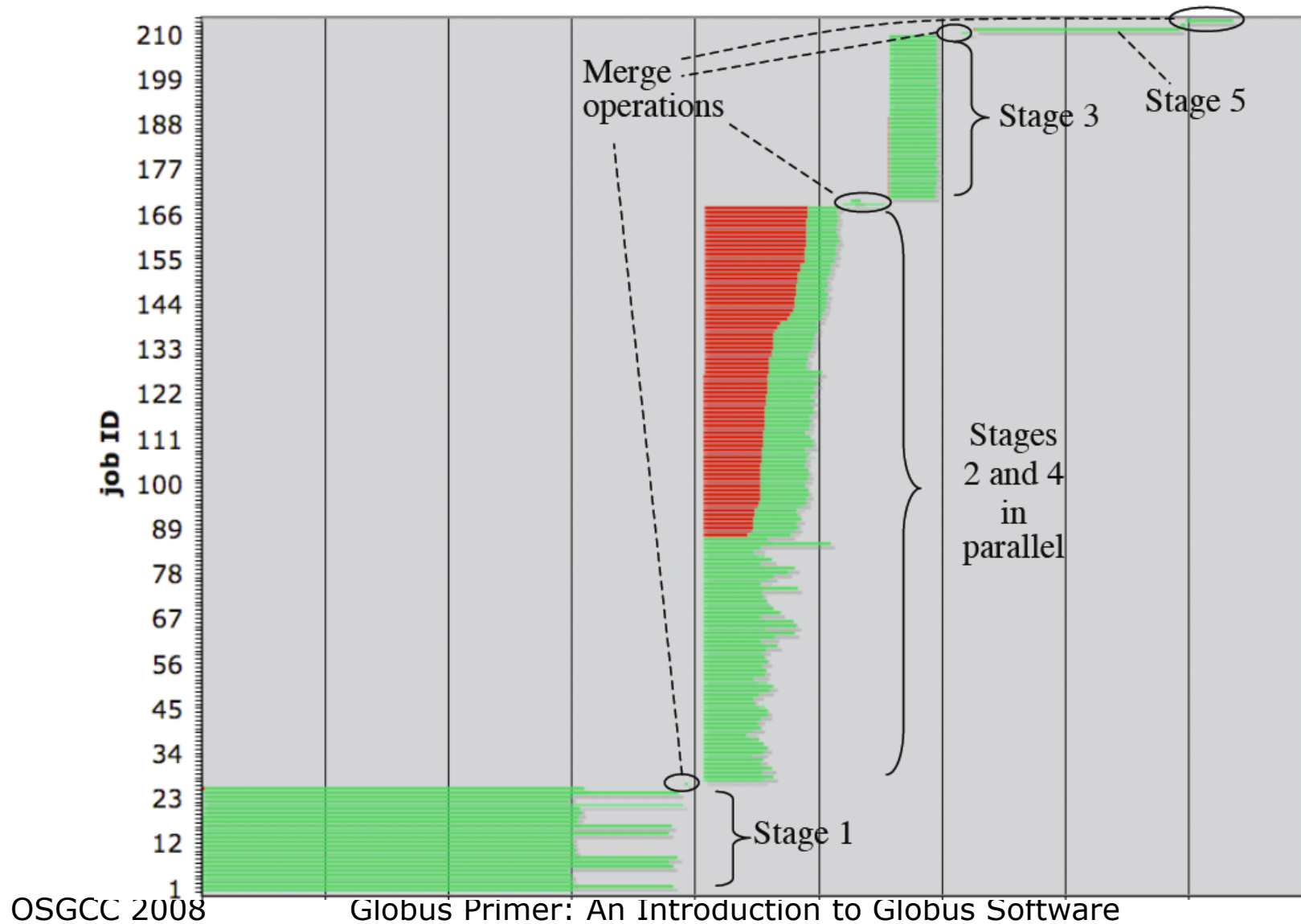
Moral Hazard SwiftScript Code Excerpts

```
// A second atomic procedure: merge
(file mergeSolutions[]) econMerge (file merging[]) {
    app{
        econMerge @filenames(mergeSolutions) @filenames(merging);
    }
}

// We define the stage one procedure—a compound procedure
(file solutions[]) stageOne (file inputData[], file prevResults[]) {
    file script<"scripts/interim.m">;
    int batch_size = 26;
    int batch_range = [0:25];
    string inputName = "IRRELEVANT";
    string outputName = "stageOneSolverOutput";
    // The foreach statement specifies that the calls can be performed concurrently
    foreach i in batch_range {
        int position = i*batch_size;
        solutions[i] = moralhazard_solver(script,batch_size,position,
                                         inputName, outputName, inputData, prevResults);
    }
}

// These get used in the “main program” as follows
stageOneSolutions = StageOne(stageOneInputFiles,stageOnePrevFiles);
stageOneOutputs = econMerge(stageOneSolutions);
```

Execution on 40 Processors





Results - Moral Hazard Solver

- Performance
 - ◆ Original run time: ~2 hrs
 - ◆ Swift run time: ~28 min
 - ◆ Depending on the stage structure, speedup up to 10x, or slowdown (because of overhead)
 - ◆ Only used one grid site (UC), on multiple sites could get better performance
- Execution has been automated
 - ◆ Human labor greatly reduced
 - ◆ Separation of human concerns (science code, system operation, task management)
 - ◆ Easy to repeat, modify & rerun, etc.



Other Applications

Application	#Jobs/computation	Levels
ATLAS* HEP Event Simulation	500K	1
fMRI DBIC* AIRSN Image Processing	100s	12
FOAM Ocean/Atmosphere Model	2000 (core app runs 250 8-CPU jobs)	3
GADU* Genomics: (14 million seq. analyzed)	40K	4
HNL fMRI Aphasia Study	500	4
NVO/NASA* Photorealistic Montage/Morphology	1000s	16
QuarkNet/I2U2* Physics Science Education	10s	3-6
RadCAD* Radiology Classifier Training	1000s	5
SIDGrid EEG Wavelet Proc, Gaze Analysis, ...	100s	20
SDSS* Coadd, Cluster Search	40K, 500K	2, 8



Globus has...

- Modular architecture
- Well-defined APIs
- Embeddable libraries
- Web service interfaces
- Globus-enabled frameworks for MPI, RPC, parallel jobs, etc.
- A very experienced support team
- Globus support on national infrastructure

Globus doesn't have...

- Your application already Grid-enabled
- A tool to automatically adapt your code
- Domain-specific frameworks



Other Grid-enabling Paths

- MPIg can run MPI applications on Grid infrastructure with little or no code change
 - ◆ Performance optimization is another story...
- Condor-G can submit tasks to GRAM2, GRAM4, Condor, etc.
- MyCluster can construct a virtual cluster out of several GRAM-accessible resources
- NinfG can run RPC applications on Grid infrastructure without even recompiling
- Introduce and gRAVI can build a Web service interface for your code and get it running on a GRAM-accessible resource so that others can invoke your code via WS

Review



THE UNIVERSITY OF
CHICAGO



Using Globus to Locate Services

- An application for a general-purpose Grid infrastructure (TeraGrid)
 - ◆ Provides a system-wide view for users in spite of independent resource providers
- MDS4 index services, info providers
 - ◆ Distributed publishing, central indexing
 - ◆ Supply your own schema or use a standard
 - ◆ Many ways to access/display data
 - ◆ Power of web services



Using Globus to Share Data

- An application for a domain-specific Grid infrastructure (LIGO)
 - ◆ Subscription-based replication of vital data for participants
- GridFTP, RFT, RLS/DRS, PyGlobus data replication
 - ◆ High-performance data transfers
 - ◆ Efficient replica tracking
 - ◆ Integration with domain-specific system



the globus alliance

www.globus.org

Using Globus for Massive Data Analysis

- Grid-enabling a domain-specific application
 - ◆ Harnessing national computational systems using a uniform job submission interface
- Leveraging GRAM4 support on national systems
 - ◆ Simplifies ease-of-use for heterogeneous resources
 - ◆ Web service model for tracking jobs
 - ◆ Integrated file stage-in, stage-out



Using Globus to Scale an Application

- A general-purpose framework for Grid-enabling applications
 - ◆ Significantly faster time-to-solution
- Swift+Falkon adapt a scientific code to run on a Grid
 - ◆ Coarse-grained parallelization
 - ◆ Automatic execution of a tedious workflow
 - ◆ Access to a shared cluster for more CPU time



Common Threads

- Each case study was a specific example
- In each case, users brought many things to the table besides Globus
 - ◆ Application code, computing systems, local resource managers, information providers, metadata, mass storage systems, etc.
- The Globus platform enables many scenarios
 - ◆ Additional known examples mentioned in each section



Status From 20,000 Feet

- Science 2.0 (Service Oriented Science) is a work in progress
 - ◆ Early adopters are banging away at it
 - ◆ Results from early applications are compelling, even transformational
 - ◆ It's only as we build, deploy, and use these systems that we identify the key capabilities
 - ◆ Abstracting these capabilities and implementing them in modular form is hard (very technical) work
- A growing suite of tools is available for use
 - ◆ http://www.globus.org/grid_software/
- We have examples of successes
 - ◆ <http://www.globus.org/solutions/>