Introduction to Grid Computing

Jennifer M. Schopf
UK National eScience Centre
Argonne National Lab

Overview and Outline

- What is a Grid
  - And what is not a Grid
- History
- Globus Toolkit and Standards
- Grid 2003 – an example application

What is a Grid

- Resource sharing
  - Computers, storage, sensors, networks, …
  - Sharing always conditional: issues of trust, policy, negotiation, payment, …
- Coordinated problem solving
  - Beyond client-server: distributed data analysis, computation, collaboration, …
- Dynamic, multi-institutional virtual orgs
  - Community overlays on classic org structures
  - Large or small, static or dynamic

Not A New Idea

- Late 70's – Networked operating systems
- Late 80's – Distributed operating system
- Early 90's – Heterogeneous computing
- Mid 90's - Metacomputing
- Then the "Grid" – Foster and Kesselman, 1999
- Also called parallel distributed computing

Why is this hard / different?

- Lack of central control
  - Where things run
  - When they run
- Shared resources
  - Contention, variability
- Communication
  - Different sites implies different sys admins, users, institutional goals, and often “strong personalities”

So why do it?

- Computations that need to be done with a time limit
- Data that can’t fit on one site
- Data owned by multiple sites
- Applications that need to be run bigger, faster, more

Grids Computing isn't just Distributed Computing

- Generally Client/Server communication
- Example: in a business
  - User interface processing – PC
  - Business processing is done in a remote computer
  - Database access/proc - another computer that provides centralized access for many business processes.
- Not a Grid:
  - All resources under central control
  - One (business) administrative domain
  - Limited different functions

Grid Computing isn't just a Distributed OS

- eg. Amoeba, Vrije Universiteit, Amsterdam
  - Users effectively log into the system as a whole, and not to a specific machine.
  - System, not the user, decides the best place to run a program
  - Single, system wide file system
  - No concept of file transfer, uploading or downloading from servers, or mounting remote file systems
- Not a Grid:
  - Complete control at a very low level over all (homogenous) resources
  - Over a LAN because of network BW constraints

Grid Computing Isn't Just Cluster Computing

- eg. Beowulf Clusters, Sterling and Becker
  - Commodity off-the-shelf personal computers
  - Interconnected LAN (Ethernet)
  - Running programs written for parallel processing
  - Multiple storage devices
  - To outside world this appears as a single system
- Not a Grid
  - Single system image
  - Central point of control
  - Single administrative domain

Grid Computing Isn't Just SETI@home

- SETI@Home, and other BOINC applications
  - Harness the power of 100,000's of computers
  - Download a small program, run when idle
  - Results (small file) are uploaded periodically
  - Central database for handing out new data sets
- Not a Grid
  - Limited functions
  - No cooperation

History

In the early 90s, Ian Foster (ANL, U-C) and Carl Kesselman (USC-ISI) enjoyed helping scientists apply distributed computing.

- Opportunities seemed ripe for the picking.
- *Application* of technology always uncovers new and interesting requirements.
- Science is cool!
- Big/Innovative science is even cooler!

What Kinds of Applications?

- Computation intensive
  - Interactive simulation (climate modeling)
  - Very large-scale simulation and analysis (galaxy formation, gravity waves, battlefield simulation)
  - Engineering (parameter studies, linked component models)
- Data intensive
  - Experimental data analysis (high-energy physics)
  - Image and sensor analysis (astronomy, climate study, ecology)
- Distributed collaboration
  - Online instrumentation (microscopes, x-ray devices, etc.)
  - Remote visualization (climate studies, biology)
  - Engineering (large-scale structural testing, chemical engineering)
- In all cases, the problems were *big* enough that they required people in several organization to *collaborate* and *share* computing *resources*, data, instruments.

What Types of Problems?

While helping to build/integrate a diverse range of applications, the same problems kept showing up over and over again.

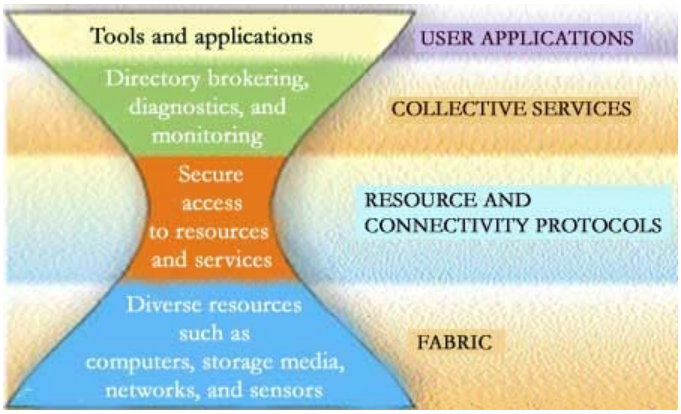
- Too hard to keep track of authentication data (ID/password) across institutions
- Too hard to monitor system and application status across institutions
- Too many ways to submit jobs
- Too many ways to store & access files and data
- Too many ways to keep track of data
- Too easy to leave “dangling” resources lying around (robustness)

What Was Needed

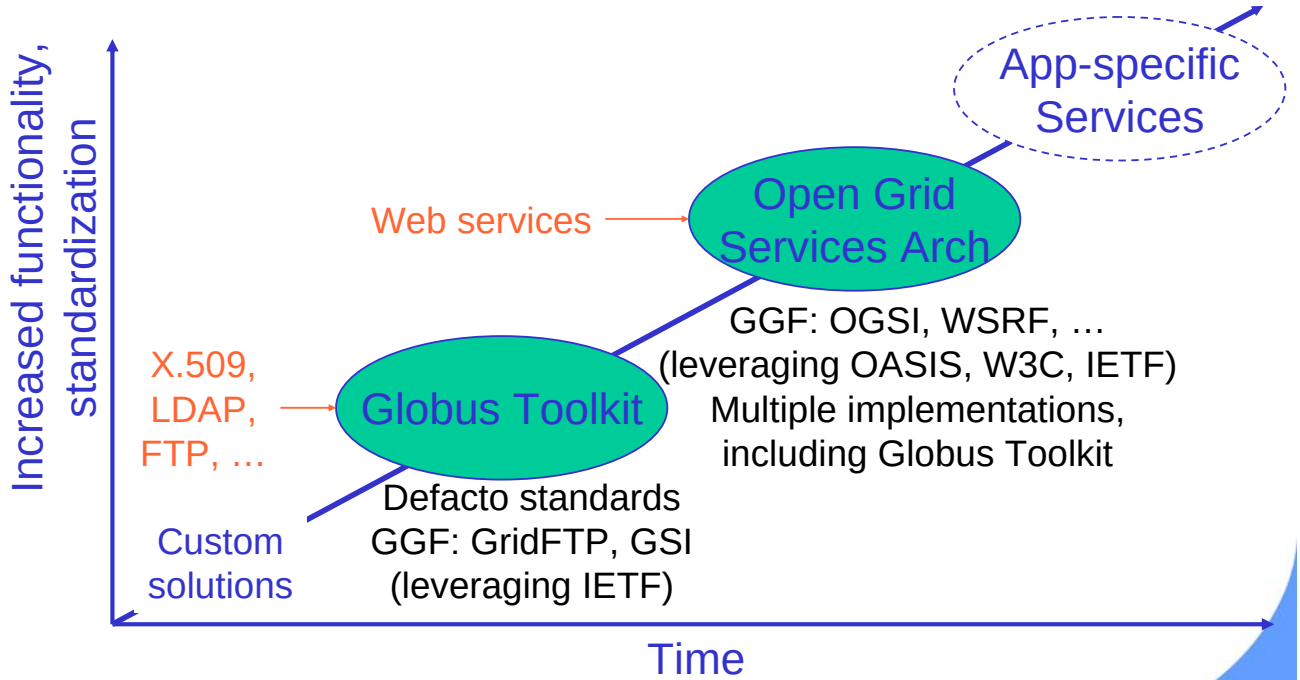
- Solutions to common problems
- Way to address heterogeniety
- Way to use standards- or to help push standards forward
  - Without standards we can’t have interoperability
- Globus Toolkit and Unicore both came up with ways to do this

With Grid Computing – Forget Homogeneity!

- Trying to force homogeneity on users is futile. Everyone has their own preferences, sometimes even *dogma*.
- The Internet provides the model…



Evolution of the Grid



Service-Oriented Architecture

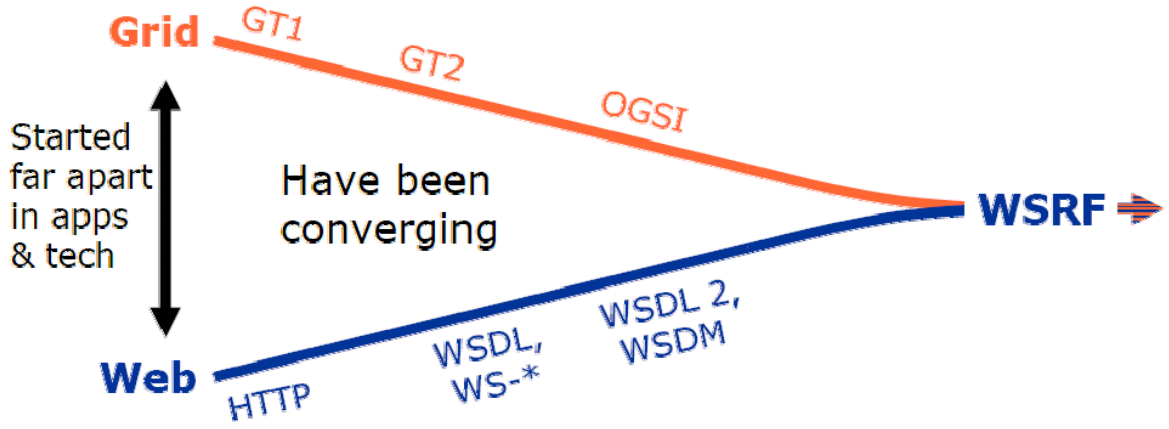
- Idea is simple (and old)
  - Define remote activities in terms of interface and behavior, not implementation
- Devil is in the details
  - How to describe, discover, access, various type of service (semantically & practically)
- Latest instantiation = Web services
  - ✓ Broad adoption, flexible XML-based model
  - ✓ WSDL, SOAP, WS-Security
  - × Interfaces still being defined to date
  - × Performance challenges

the globus alliance
www.globus.org
National e-Science Centre

Open Grid Services Architecture

- Define a service-oriented architecture…
  - the key to effective virtualization
- …to address vital Grid requirements
  - AKA utility, on-demand, system management, collaborative computing, etc.
- …building on Web service standards.
  - extending those standards when needed

Grid and Web Services Convergence



The definition of WSRF means that the Grid and Web services communities can move forward on a common base.

the globus alliance
www.globus.org
National e-Science Centre

Overview and Outline

- What is a Grid
  - And what is not a Grid
- History
- Globus Toolkit and Standards
  - Background
  - Security
  - Data Management
  - Resource Management
  - Monitoring
- Grid 2003 – an Example application

Globus Is "Standard Plumbing" for the Grid

- *Not* turnkey solutions, but *building blocks* and *tools* for application developers and system integrators.
  - Some components (e.g., file transfer) go farther than others (e.g., remote job submission) toward end-user relevance.
- Since these solutions exist and others are already using them (and they're free), it's easier to reuse than to reinvent.
  - And compatibility with other Grid systems comes for free!

Leveraging Existing and Proposed Standards

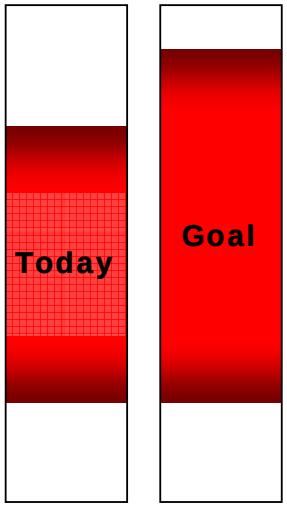
- SSL/TLS v1 (from OpenSSL) (IETF)
- LDAP v3 (from OpenLDAP) (IETF)
- X.509 Proxy Certificates (IETF)
- GridFTP v1.0 (GGF)
- OGSI v1.0 (GGF)
- And others on the road to standardization:

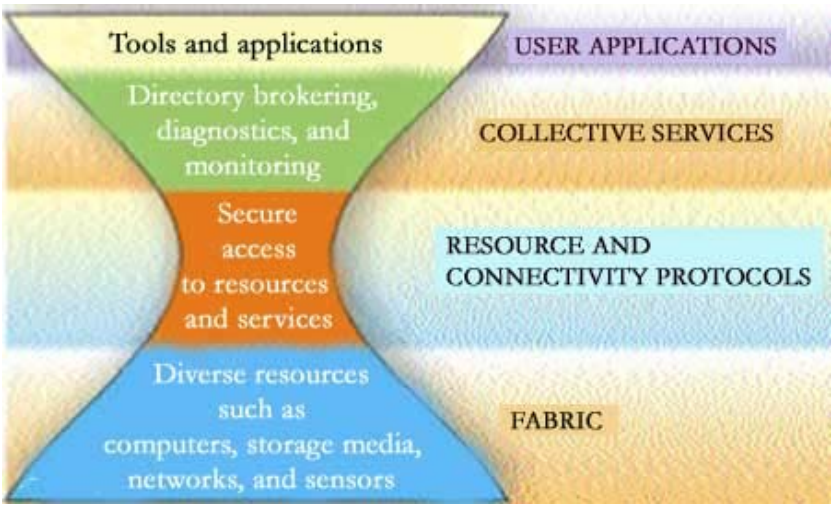
  WSRF (GGF, OASIS), DAI, WS-Agreement, WSDL 2.0, WSDM, SAML, XACML

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 heterogenity* for application developers.
  - We aspire to include more “vertical solutions” in future versions.
- 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 Does the Globus Toolkit Cover?



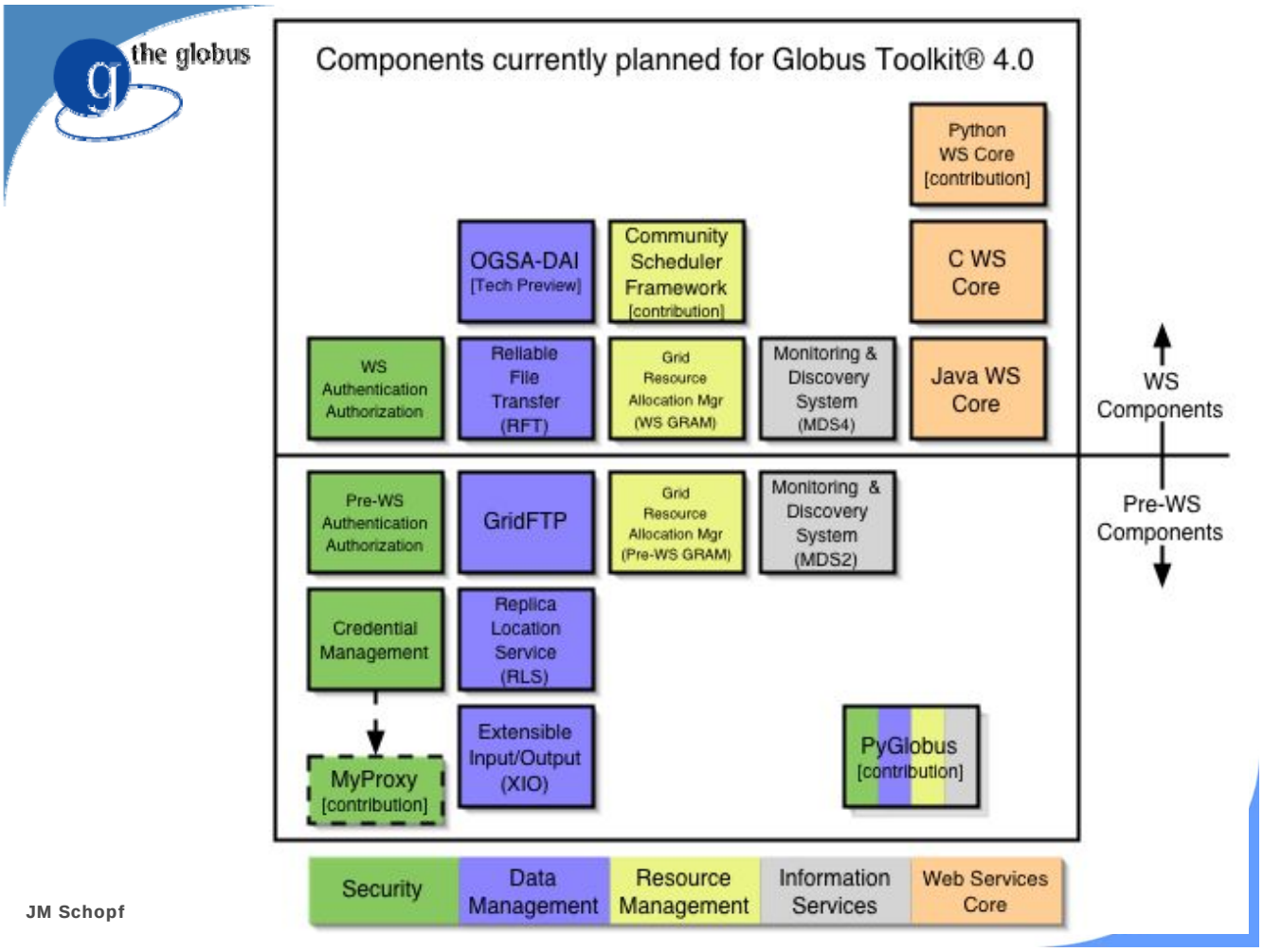


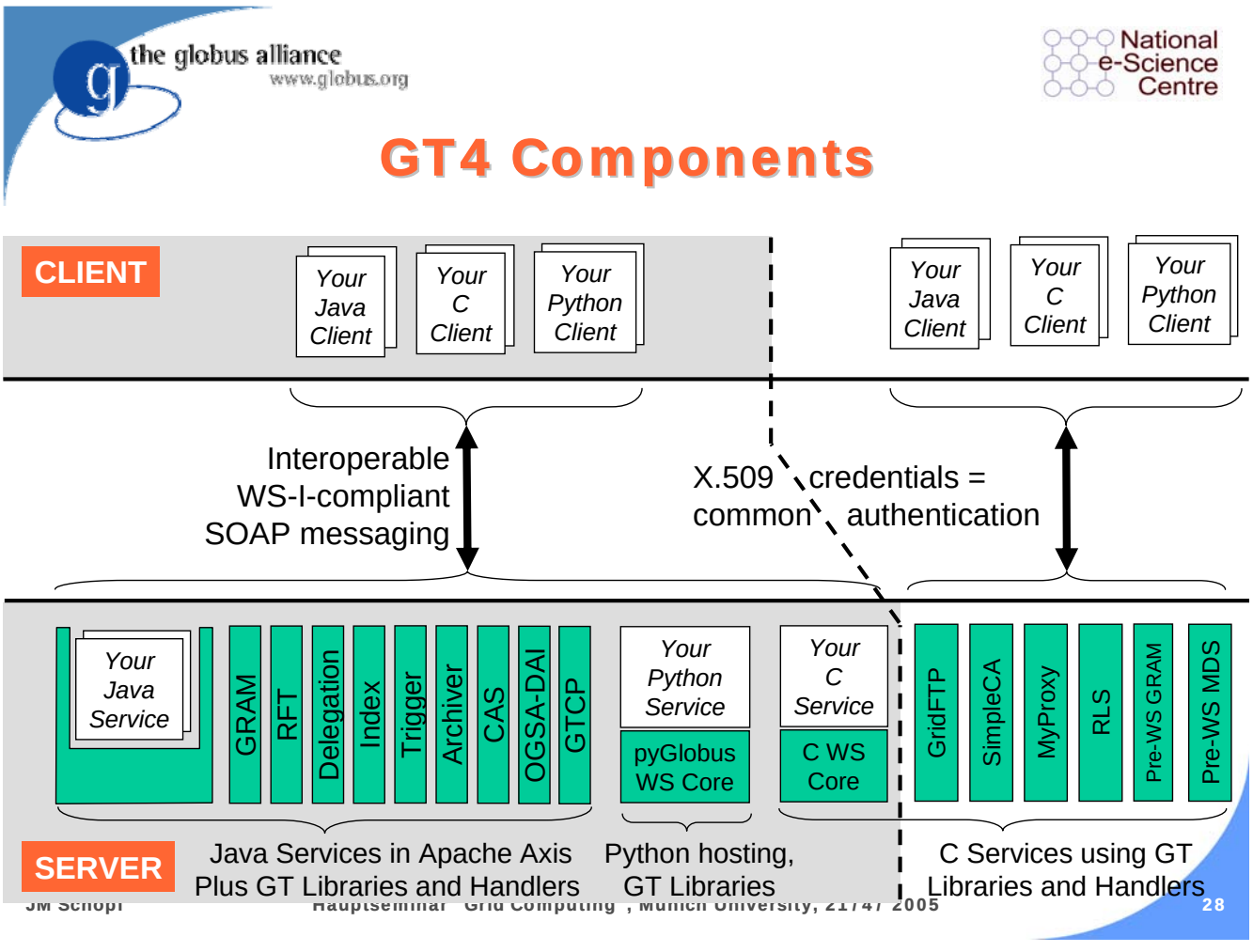
Areas of Competence

- "Connectivity Layer" Solutions
  - Service Management (WSRF)
  - Monitoring/Discovery (WSRF and MDS)
  - Security (GSI and WS-Security)
  - Communication (XIO)
- "Resource Layer" Solutions
  - Computing / Processing Power (GRAM)
  - Data Access/Movement (GridFTP, OGSA-DAI)
- "Collective Layer" Solutions
  - Data Management (RLS, MCS, OGSA-DAI)
  - Monitoring/Discovery (MDS)
  - Security (CAS)

What Is 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
- Note: GT3 and GT4 releases include both WS and pre-WS components!

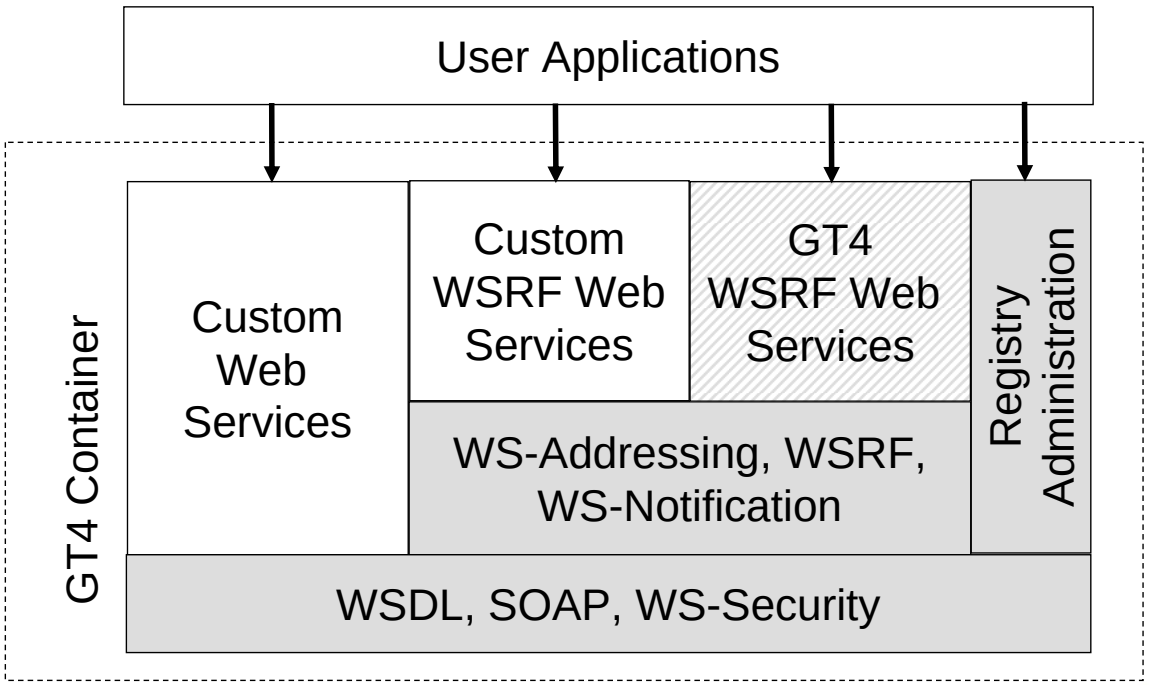

the globus
Components currently planned for Globus Toolkit® 4.0
Python WS Core [contribution]
OGSA-DAI [Tech Preview]
Community Scheduler Framework [contribution]
C WS Core
WS Authentication Authorization
Reliable File Transfer (RFT)
Grid Resource Allocation Mgr (WS GRAM)
Monitoring & Discovery System (MDS4)
Java WS Core
WS Components
Pre-WS Authentication Authorization
GridFTP
Grid Resource Allocation Mgr (Pre-WS GRAM)
Monitoring & Discovery System (MDS2)
Pre-WS Components
Credential Management
Replica Location Service (RLS)
Extensible Input/Output (XIO)
PyGlobus [contribution]
MyProxy [contribution]
Security
Data Management
Resource Management
Information Services
Web Services Core
JM Schopf


the globus alliance
www.globus.org
National e-Science Centre
GT4 Components
CLIENT
Your Java Client
Your C Client
Your Python Client
Your Java Client
Your C Client
Your Python Client
Interoperable WS-I-compliant SOAP messaging
X.509 credentials = common authentication
Your Java Service
GRAM
RFT
Delegation
Index
Trigger
Archiver
CAS
OGSA-DAI
GTCP
Your Python Service
pyGlobus WS Core
Your C Service
C WS Core
GridFTP
SimpleCA
MyProxy
RLS
Pre-WS GRAM
Pre-WS MDS
SERVER
Java Services in Apache Axis Plus GT Libraries and Handlers
Python hosting, GT Libraries
C Services using GT Libraries and Handlers
JM Schopf
Hauptseminar "Grid Computing", Munich University, 21/47/2005
28

GT4 Web Services Core

- Supports both Globus services (GRAM, RFT, Delegation, etc.) & user-developed services
- Redesign to enhance scalability, modularity, performance, usability
- Leverages existing WS standards
  - WS-I Basic Profile: WSDL, SOAP, etc.
  - WS-Security, WS-Addressing
- Adds support for emerging WS standards
  - WS-Resource Framework, WS-Notification
- Java, Python, & C hosting environments

GT4 Web Services Core



Open Source / Open Standards

- WSRF developed in collaboration with IBM
  - Currently in OASIS process
- Contributions to Apache for
  - WS-Security
  - WS-Addressing
  - Axis
  - Apollo (WSRF)
  - Hermes (WS-Notification)

Overview and Outline

- What is a Grid
  - And what is not a Grid
- History
- Globus Toolkit and Standards
  - Background
  - Security
  - Data Management
  - Resource Management
  - Monitoring
- Grid 2003 – an Example application

Why Grid Security is Hard

- 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

Basic Grid Security Mechanisms

- Basic Grid authentication and authorization mechanisms come in two flavors.
  - Pre-Web services
  - Web services
- Both are included in the Globus Toolkit, and both provide vital security features.
  - Grid-wide identities implemented as PKI certificates
  - Transport-level and message-level authentication
  - Ability to delegate credentials to agents
  - Ability to map between Grid & local identities
  - Local security administration & enforcement
  - Single sign-on support implemented as “proxies”
  - A “plug in” framework for authorization decisions

Basic Grid Security Mechanisms

- Basic security mechanisms are provided as libraries/classes and APIs.
  - Integrated with other GT tools and services
  - Integrated with many Grid community tools and services (and applications & systems)
- A few stand-alone tools are also included.

A Cautionary Note

- Grid security mechanisms are tedious to set up.
  - If exposed to users, hand-holding is usually required.
  - These mechanisms can be *hidden entirely* from end users, but still used behind the scenes.
- These mechanisms exist for good reasons.
  - Many useful things can be done without Grid security.
  - It is unlikely that an ambitious project could go into production operation without security like this.
  - Most successful projects end up using Grid security, but using it in ways that end users don't see much.

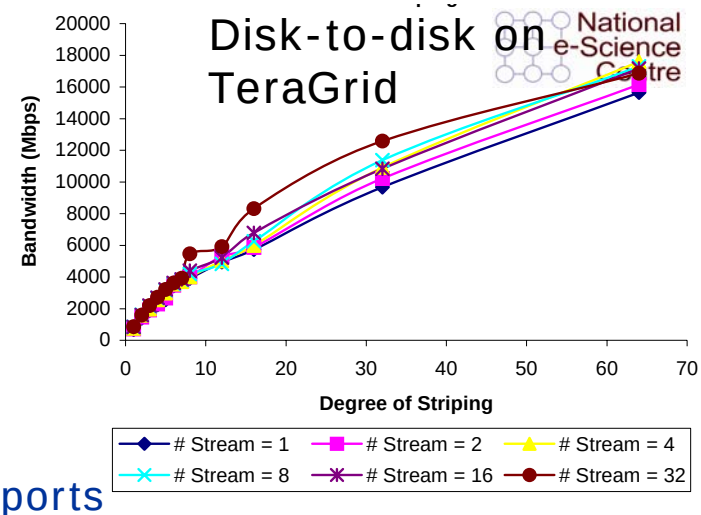
GT4 Data Management

- Stage large data to/from nodes
- Replicate data for performance & reliability
- Locate data of interest
- Provide access to diverse data sources
  - File systems, parallel file systems, hierarchical storage (GridFTP)
  - Databases (OGSA DAI)

GT4 Data Functions

- Find your data: **Replica Location Service**
  - Managing ~40M files in production settings
- Move/access your data: **GridFTP**, **RFT**
  - High-performance striped data movement
    - > 27 Gbit/s memory-to-memory on a 30 Gbit/s link (90% utilization) with 32 IBM TeraGrid nodes.
    - > 17.5 Gbit/s disk-to-disk limited by the storage system
    - > Reliable movement of 120,000 files (so far)
- Couple data & execution management
  - GRAM uses GridFTP and RFT for staging

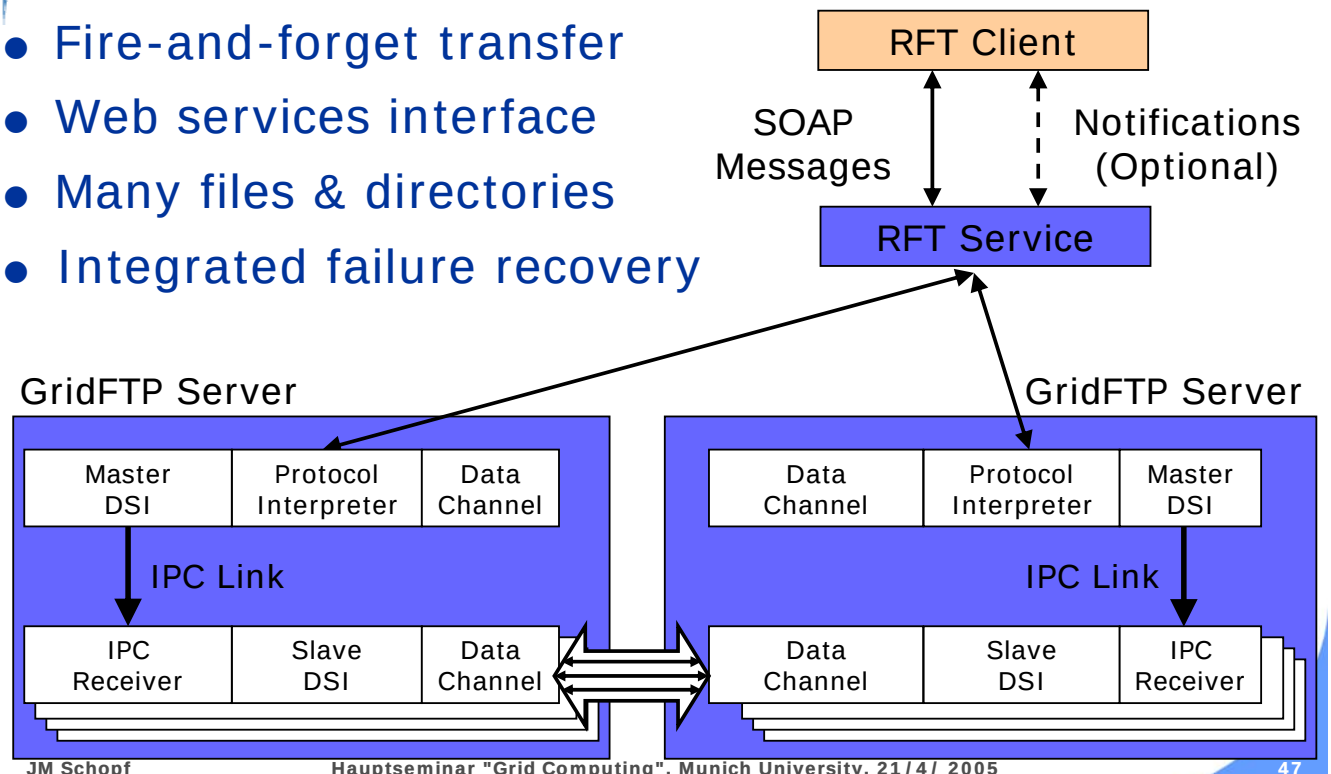
GridFTP in GT4



- 100% Globus code
  - No licensing issues
  - Stable, extensible
- IPv6 Support
- XIO for different transports
- Striping → multi-Gb/sec wide area transport
- Pluggable
  - Front-end: e.g., future WS control channel
  - Back-end: e.g., HPSS, cluster file systems
  - Transfer: e.g., UDP, NetBLT transport

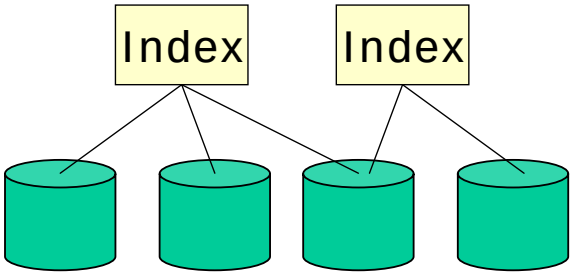
Reliable File Transfer: Third Party Transfer

- Fire-and-forget transfer
- Web services interface
- Many files & directories
- Integrated failure recovery



Replica Location Service

- Identify location of files via logical to physical name map
- Distributed indexing of names, fault tolerant update protocols
- GT4 version scalable & stable
- Managing ~40 million files across ~10 sites



Local DB	Update send (secs)	Bloom filter (secs)	Bloom filter (bits)
10K	<1	2	1 M
1 M	2	24	10 M
5 M	7	175	50 M

OGSA-DAI

- Grid Interfaces to Databases
  - Data access
    - > Relational & XML Databases, semi-structured files
  - Data integration
    - > Multiple data delivery mechanisms, data translation
- Extensible & Efficient framework
  - Request documents contain multiple tasks
    - > A task = execution of an activity
    - > Group work to enable efficient operation
  - Extensible set of activities
    - > > 30 predefined, framework for writing your own
  - Moves computation to data
  - Pipelined and streaming evaluation
  - Concurrent task evaluation

Overview and Outline

- What is a Grid
  - And what is not a Grid
- History
- Globus Toolkit and Standards
  - Background
  - Security
  - Data Management
  - Resource Management
  - Monitoring
- Grid 2003 – an Example application

The Resource Management Challenge

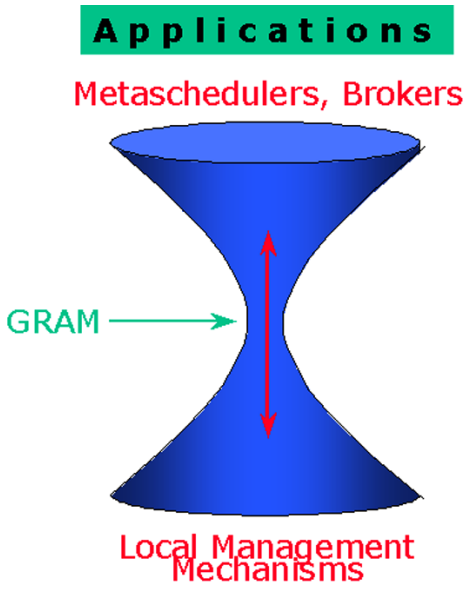
- Enabling secure, controlled remote access to heterogeneous computational resources and management of remote computation
  - Authentication and authorization
  - Resource discovery & characterization
  - Reservation and allocation
  - Computation monitoring and control
- Addressed by a set of protocols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

Execution Management (GRAM)

- Common WS interface to schedulers
  - Unix, Condor, LSF, PBS, SGE, ...
- More generally: interface for process execution management
  - Set up execution environment
  - Stage data
  - Monitor & manage lifecycle
  - Kill it, clean up

GRAM - Basic Job Submission and Control Service

- A uniform service interface for remote job submission and control
  - 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



GT4 GRAM

- 2nd-generation WS implementation
  - optimized for performance, stability, scalability
- Streamlined critical path
  - Use only what you need
- Flexible credential management
  - Credential cache & delegation service
- GridFTP & RFT used for data operations
  - Data staging & streaming output
  - Eliminates redundant GASS code
- Single and multi-job support

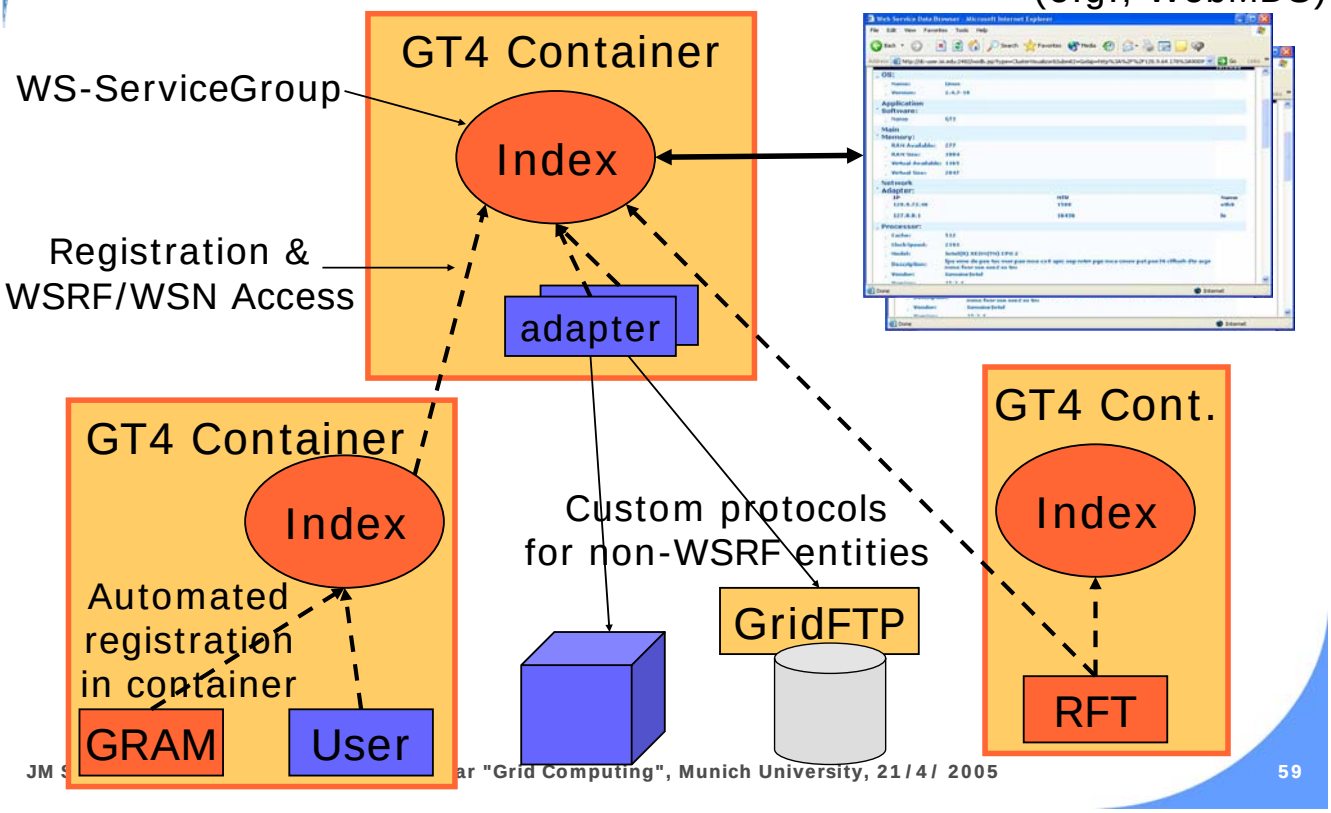
Monitoring and Discovery Challenges

- Grid Information Service
- Requirements and characteristics
  - Uniform, flexible access to information
  - Scalable, efficient access to dynamic data
  - Access to multiple information sources
  - Decentralized maintenance
  - Secure information provision

Monitoring and Discovery

- "Every service should be monitorable and discoverable using common mechanisms"
  - WSRF/WSN provides those mechanisms
- A common aggregator framework for collecting information from services, thus:
  - Index Service: Registry supporting Xpath queries, with caching
  - Trigger Service: Performs action on condition
- Deep integration with Globus containers & services: every GT4 service is discoverable
  - GRAM, RFT, CAS, ...

GT4 Monitoring & Discovery



MDS4 Extensibility

- Aggregator framework provides
  - Registration management
  - Collection of information from Grid Resources
  - Plug in interface for data access, collection, query, …
- WebMDS framework provides for customized display
  - XSLT transformations

The Globus Ecosystem

- **Globus components** address core issues relating to resource access, monitoring, discovery, security, data movement, etc.
  - GT4 being the latest version
- A larger **Globus ecosystem** of open source and proprietary components provide complementary components
  - A growing list of components
- These components can be combined to produce **solutions** to Grid problems
  - We're building a list of such solutions

Many Tools Build on, or Can Contribute to, GT4-Based Grids

- Condor-G, DAGman
- MPICH-G2
- GRMS
- Nimrod-G
- Ninf-G
- Open Grid Computing Env.
- Commodity Grid Toolkit
- GriPhyN Virtual Data System
- Virtual Data Toolkit
- GridXpert Synergy
- Platform Globus Toolkit
- VOMS
- PERMIS
- GT4IDE
- Sun Grid Engine
- PBS scheduler
- LSF scheduler
- GridBus
- TeraGrid CTSS
- NEES
- IBM Grid Toolbox
- ...

2005 and Beyond

- We have a solid Web services base
- We now want to build, on that base, a open source service-oriented infrastructure
  - Virtualization
  - New services for provisioning, data management, security, VO management
  - End-user tools for application development
  - Etc., etc.

How To Get Involved

- Download the software and start using it
  - **http://www.globus.org/toolkit/**
- Provide feedback
  - Join **gt4-friends@globus.org** mail list
  - File bugs at **http://bugzilla.globus.org**
- Review, critique, add to documentation
  - Globus Doc Project: **http://gdp.globus.org**
- Tell us about your GT4-related tool, service, or application
  - Email **info@globus.org**

Globus and its User Community

- How can “we” best support “you”?
  - We try to provide the best software we can
  - We use bugzilla & other community tools
  - We work to grow the set of contributors
- How can “you” best support “us”?
  - Become a contributor: of software, bug fixes, answers to questions, documentation
  - Provide us with success stories that can justify continued Globus development
  - Promote Globus within your communities

Overview and Outline

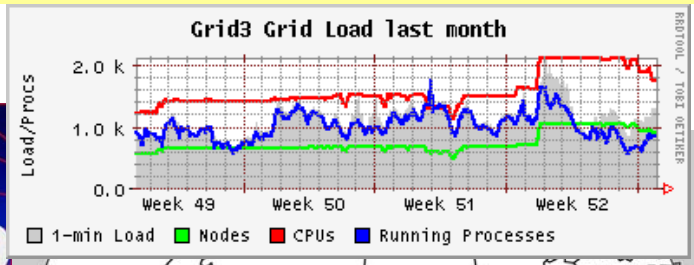
- What is a Grid
  - And what is not a Grid
- History
- Globus Toolkit and Standards
- Grid 2003 – an Example application

Using the Globus Toolkit: How it *Really* Happens

- Implementations are provided by a mix of
  - Application-specific code
  - "Off the shelf" tools and services
  - Tools and services from the Globus Toolkit
  - Tools and services from the Grid community (compatible with GT)
- Glued together by…
  - Application development
  - System integration

Grid2003: An Operational Grid

- 28 sites (2100-2800 CPUs) & growing
- 10 substantial applications + CS experiments
- Running since October 2003, still up today



http://www.ivdgl.org/grid2003

the globus alliance
www.globus.org

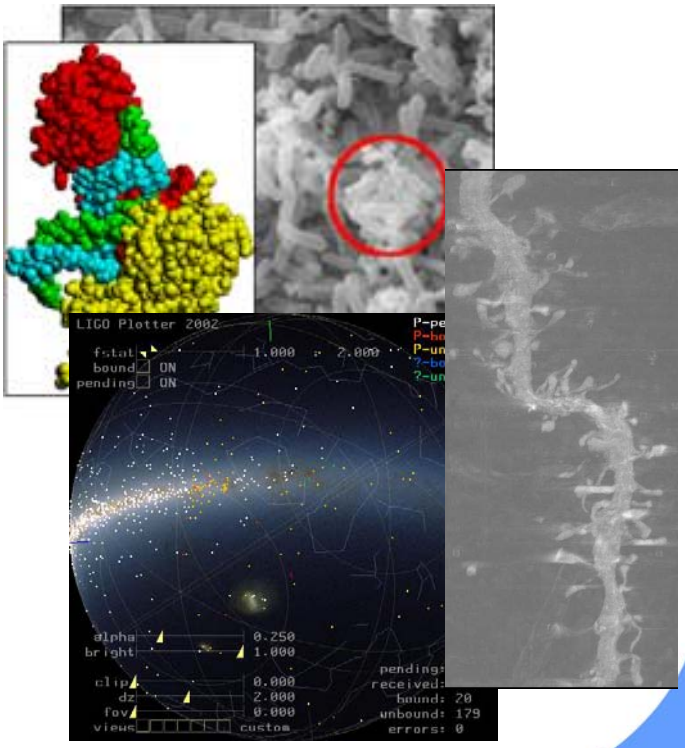
National e-Science Centre

Grid2003 Project Goals

- Ramp up U.S. Grid capabilities in anticipation of LHC experiment needs in 2005.
  - Build, deploy, and operate a working Grid.
  - Include all U.S. LHC institutions.
  - Run real scientific applications on the Grid.
  - Provide state-of-the-art monitoring services.
  - Cover non-technical issues (e.g., SLAs) as well as technical ones.
- Unite the U.S. CS and Physics projects that are aimed at support for LHC.
  - Common infrastructure
  - Joint (collaborative) work

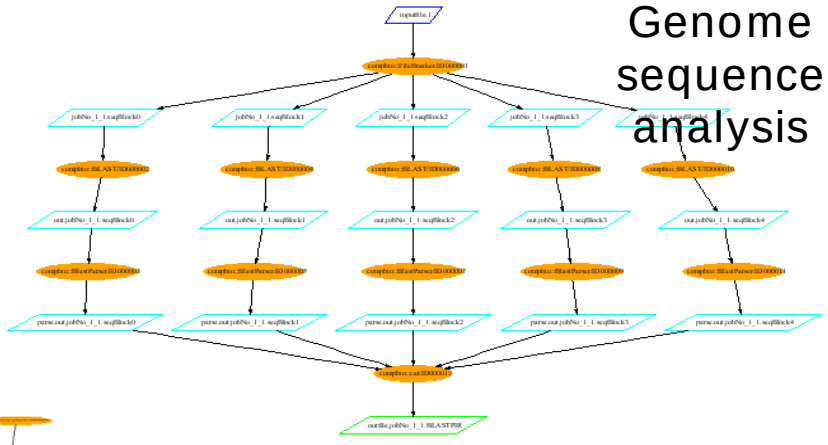
Grid2003 Applications

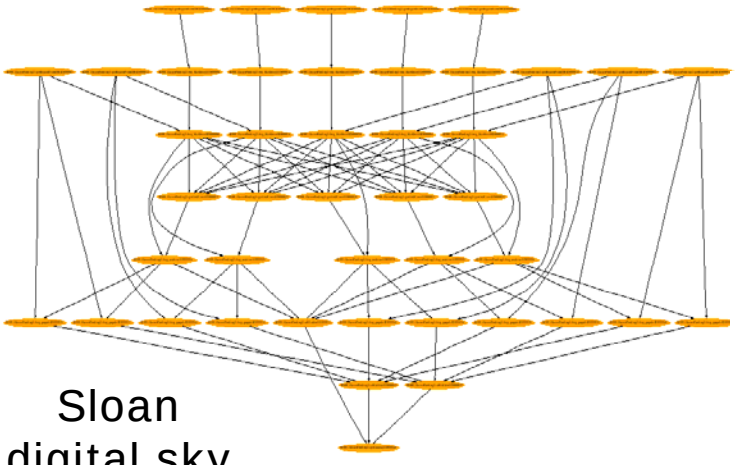
- 6 VOs, 10 Apps +CS
- CMS proton-proton collision simulation
- ATLAS proton-proton collision simulation
- LIGO gravitational wave search
- SDSS galaxy cluster detection
- ATLAS interactive analysis
- BTeV proton-antiproton collision simulation
- SnB biomolecular analysis
- GADU/Gnare genone analysis
- And more!



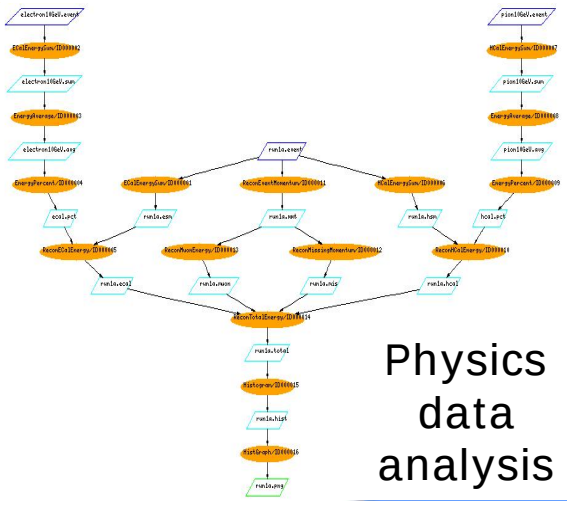
Example Grid2003 Workflows

Genome sequence analysis





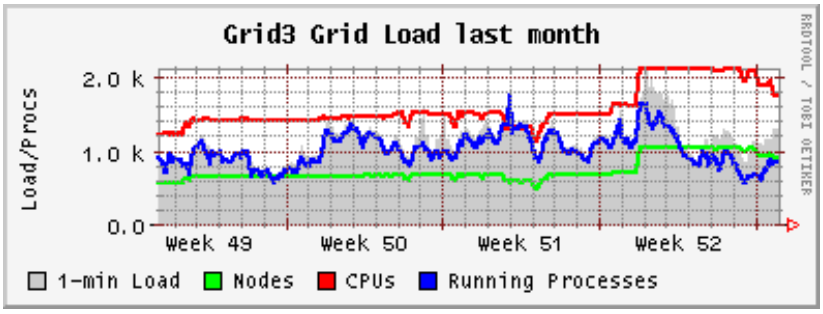
Sloan digital sky survey



Physics data analysis

Grid2003 Requirements

- General Infrastructure
- Support Multiple Virtual Organizations
- Production Infrastructure
- Standard Grid Services
- Interoperability with European LHC Sites
- Easily Deployable
- Meaningful Performance Measurements



Grid 2003 Components

- Computers & storage at 28 sites (to date)
  - 2800+ CPUs
- Uniform service environment at each site
  - Set of software that is deployed on every site
  - Pacman installation system enables installation of numerous other VDT and application services
- Global & virtual organization services
  - Certification & registration authorities, VO membership services, monitoring services
- Client-side tools for data access & analysis
  - Virtual data, execution planning, DAG management, execution management, monitoring
- IGOC: iVDGL Grid Operations Center

SW Components: Security

- GT Components
  - GSI
  - Community Authorization Service (CAS)
  - MyProxy
- Related Components
  - GSI-OpenSSH

SW Components: Job Submission

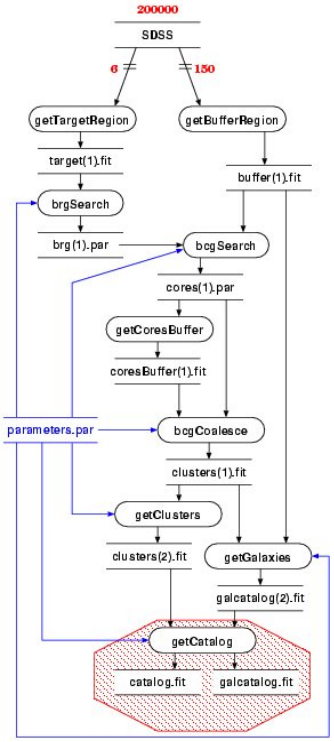
- GT components
  - pre-ws GRAM
  - Condor-G
- Related components
  - Chimera – Virtual Data Management
  - Pegasus – Workflow Management

CondorG

- The Condor project has produced a “helper front-end” to GRAM
  - Managing sets of subtasks
  - Reliable front-end to GRAM to manage computational resources
- Note: this is not Condor which promotes high-throughput computing, and use of idle resources

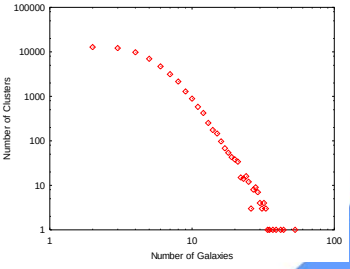
Chimera “Virtual Data”

- Captures both logical and physical steps in a data analysis process.
  - Transformations (logical)
  - Derivations (physical)
- Builds a catalog.
- Results can be used to “replay” analysis.
  - Generation of DAG (via Pegasus)
  - Execution on Grid
- Catalog allows introspection of analysis process.



Sloan Survey Data

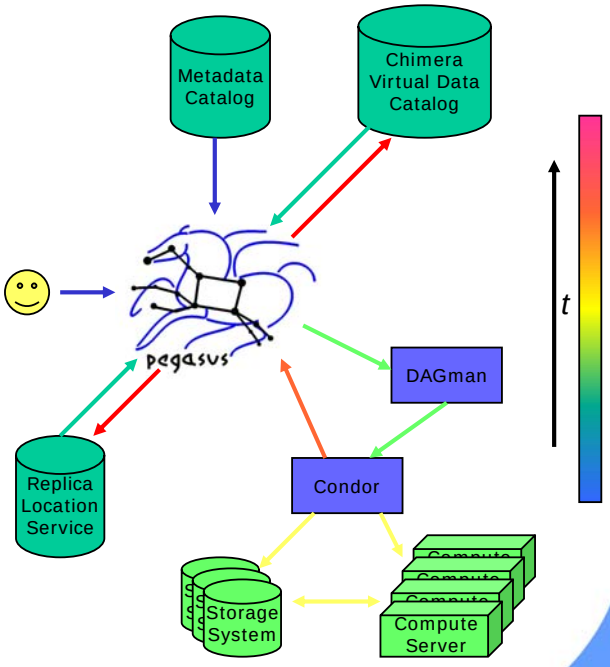
Galaxy cluster size distribution



Pegasus Workflow Transformation

Converts Abstract Workflow (AW) into Concrete Workflow (CW).

- Uses Metadata to convert user request to logical data sources
- Obtains AW from Chimera
- Uses replication data to locate physical files
- Delivers CW to DAGman
- Executes
- Publishes new replication and derivation data in RLS and Chimera (optional)



SW Components: Data Tools

- GT Components
  - GridFTP (old)
  - Replica Location Service (RLS)
- Related components
  - ISI Metadata Catalog Service

MCS - Metadata Catalog Service

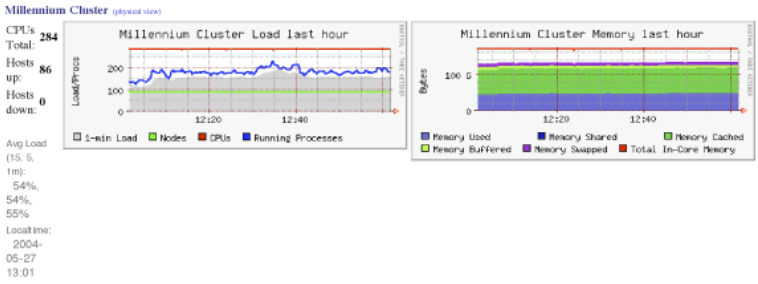
- A stand-alone metadata catalog service
  - Stores system-defined and user-defined attributes for logical files/objects
  - Supports manipulation and query
- Integrated with OGSA-DAI
  - OGSA-DAI provides metadata storage
  - When run with OGSA-DAI, basic Grid authentication mechanisms are available

SW Components: Monitoring

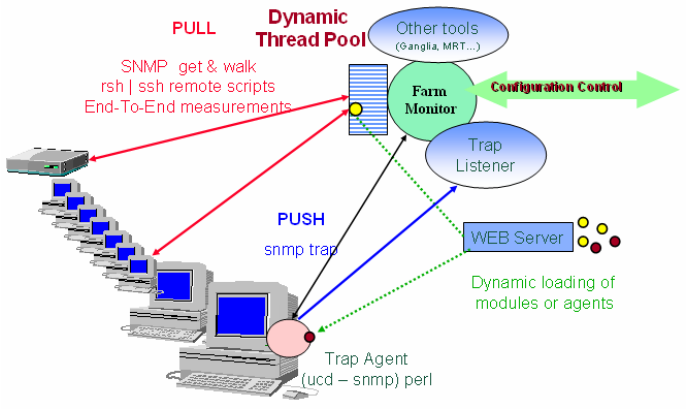
- GT components
  - MDS2 (basically equivalent to MDS4 index server)
- Related components
  - 8 other tools including Ganglia, MonALISA, home grown add-ons

Ganglia Cluster Monitor

- Ganglia is a toolkit for monitoring clusters and aggregations of clusters (hierarchically).
- Ganglia collects system status information and makes it available via a web interface.
- Ganglia status can be subscribed to and aggregated across multiple systems.
- Integrating Ganglia with MDS services results in status information provided in the proposed standard GLUE schema, popular in international Grid collaborations.



MonALISA



- Supports system-wide, distributed monitoring with triggers for alerts
  - Java/JINI and Web services
  - Integration with Ganglia, queuing systems, etc.
  - Client interfaces include Web and WAP
  - Flexible registration and proxy mechanisms support look-ups and firewalls

Grid2003 Operation

- All software to be deployed is integrated in the Virtual Data Toolkit (VDT) distribution.
  - Each participating institution deploys the VDT on their systems, which provides a standard set of software and configuration.
  - A core software team (GriPhyN, iVDGL) is responsible for integration and development.
- A set of centralized services (e.g., directory services, MyProxy service) is maintained Grid-wide.
- Applications are developed with VDT capabilities, architecture, and services directly in mind.

Grid2003 Metrics

Metric	Target	Achieved
Number of CPUs	400	2762 (28 sites)
Number of users	> 10	102+
Number of applications	> 4	10 (+CS)
Number of sites running concurrent apps	> 10	17
Peak number of concurrent jobs	1000	1100
Data transfer per day	> 2-3 TB	4.4 TB max

Grid2003 Summary

- Working Grid for wide set of applications
- Joint effort between application scientists, computer scientists
- Globus software as a starting point, additions from other communities as needed
- Transitioning to GT4 one component at a time

Overall Summary

- Grid computing offers a number of challenges, but also opportunity, for collaborative projects
- Open issues are changing – but that's the nature of this field
- Globus Toolkit offers a standards-based set of building blocks
- Many users are trying to make this work for them – you can too!

For More Information

- Globus Toolkit
  - www.globus.org/toolkit
- Grid EcoSystem
  - www-unix.grids-center.org/r6/ecosystem/