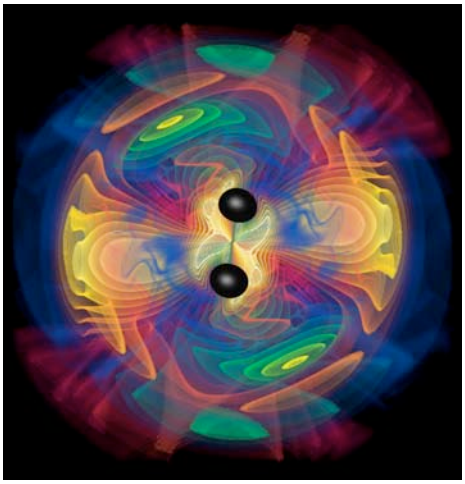
Astrophysics on Grids with GridLab and Cactus



Gabrielle Allen

gallen@cct.lsu.edu

Center for Computation & Technology, LSU

(Max Planck Institute for Gravitational Physics)

Center for Computation & Technology

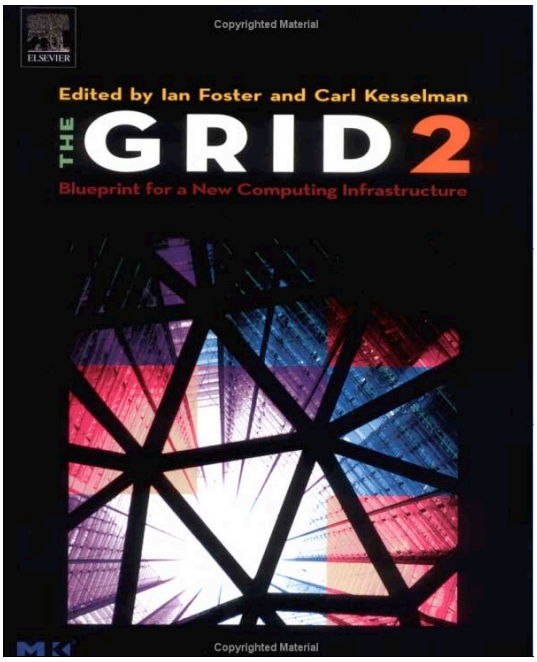
AT LOUISIANA STATE UNIVERSITY

Thanks ...

Chapter 16

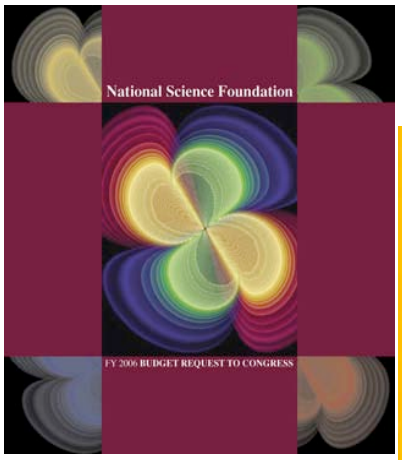
Ed Seidel & Gabrielle Allen

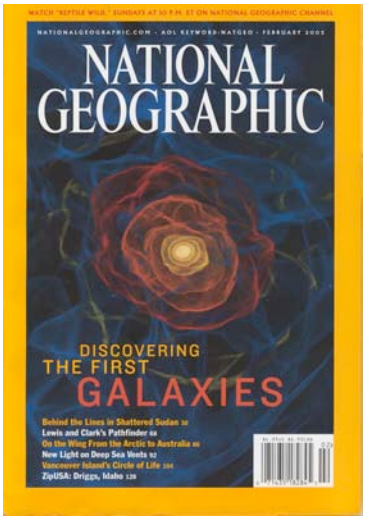
Collaborative Science: Astrophysics Requirements and Experiences



Challenge Problems in Astrophysics and Astronomy

- Cosmology
- Black Hole and Neutron Star Models
- Supernovae
- Astronomical Databases
- Gravitational Wave Data Analysis
- Need and drive HEC & Grids

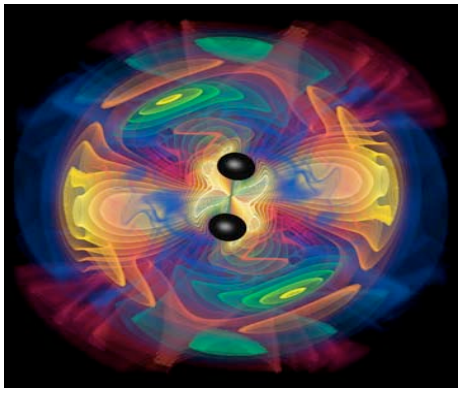




Gravitational Wave Physics

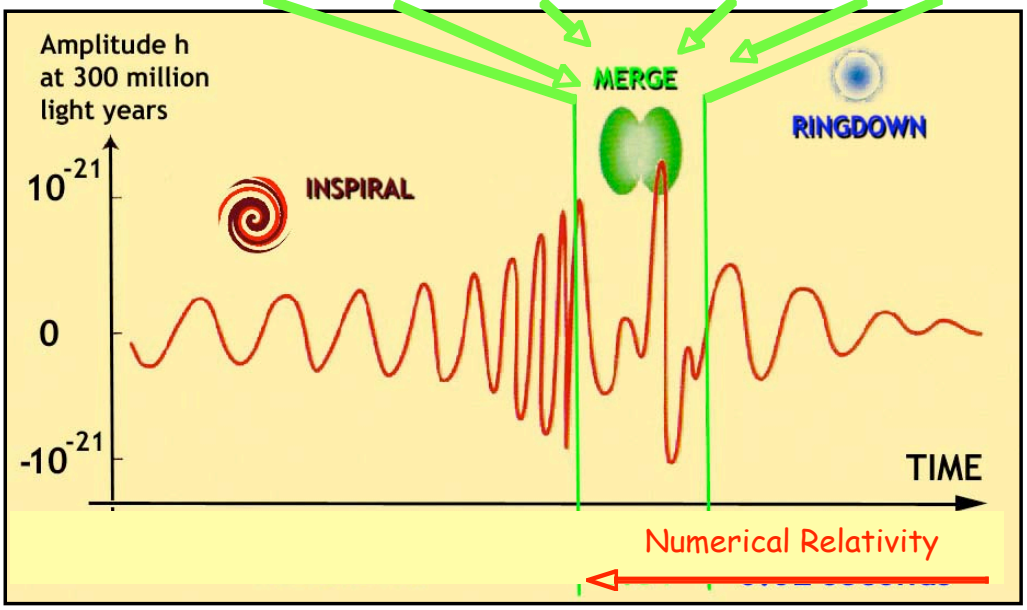
Observations

Models



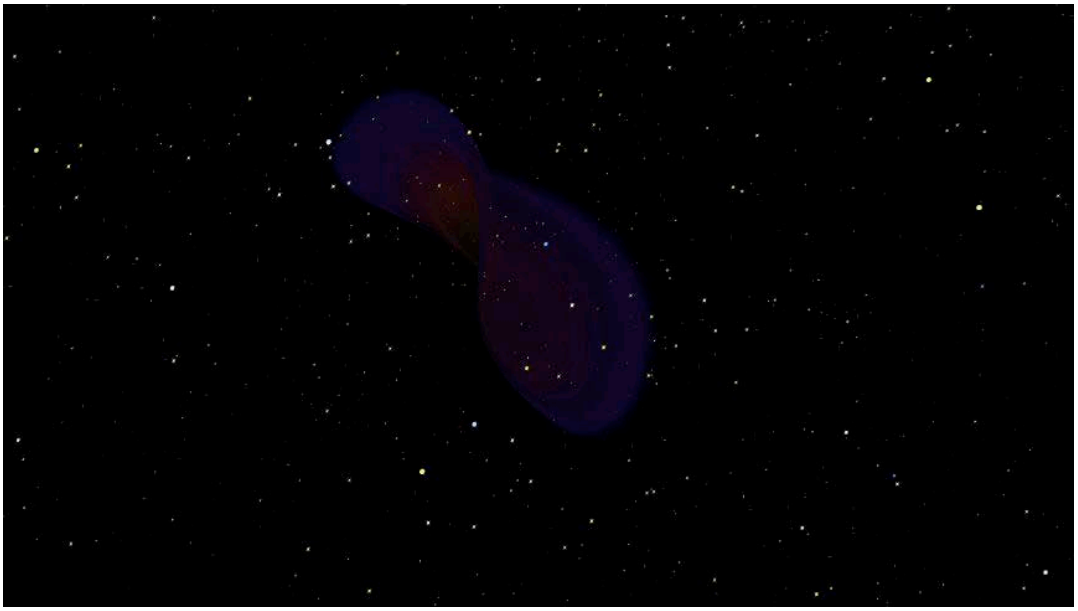
Modeling Nature

Teraflop Computation, AMR, Elliptic-Hyperbolic, ???

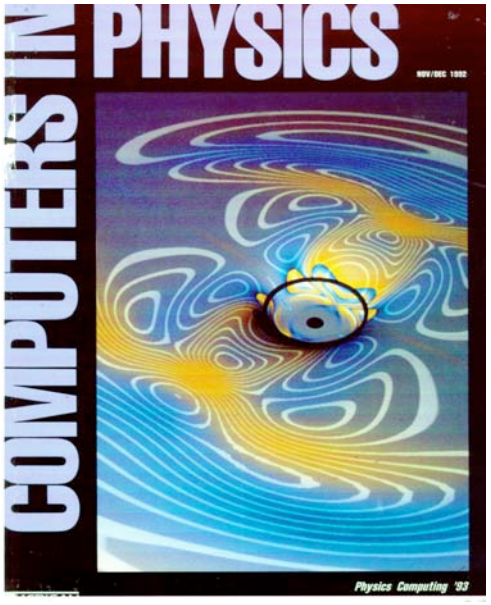


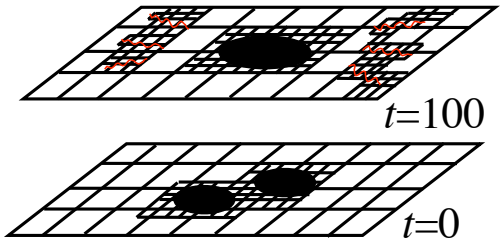
Huge Black Hole Collision Simulation

3000 frames of volume rendering TB of data



3D Numerical Relativity

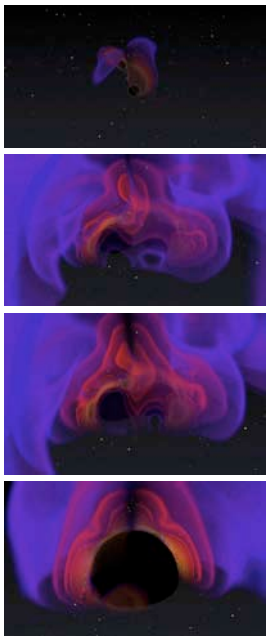




Get physicists + CS people together
Find Resource (TByte, TFlop)
Initial Data: 4 coupled nonlin. elliptics
Choose Gauge (elliptic/hyperbolic…)
Evolution
"hyperbolic" evolution
coupled with elliptic eqs.
Find Resource ….
Analysis: Interpret, Find AH, etc

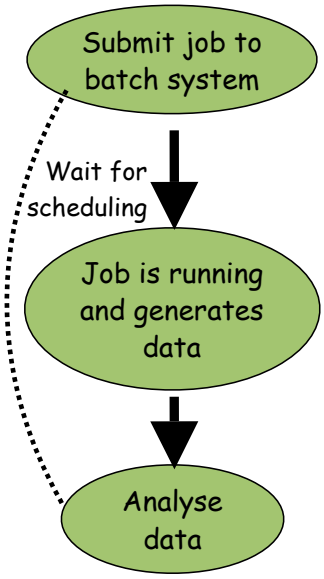
Computational Science Needs

- Requires incredible mix of technologies & expertise!
- Many scientific/engineering components
  - Physics, astrophysics, CFD, engineering,…
- Many numerical algorithm components
  - Finite difference? Finite volume? Finite elements?
  - Elliptic equations: multigrid, Krylov subspace,…
  - Mesh refinement?
- Many different computational components
  - Parallelism (HPF, MPI, PVM, ???)
  - Architecture (MPP, DSM, Vector, PC Clusters, ???)
  - I/O (generate GBs/simulation, checkpointing…)
  - Visualization of all that comes out!
- Such work cuts across many disciplines, areas of CS…



Monitoring and Steering

Shorten the traditional production cycle:

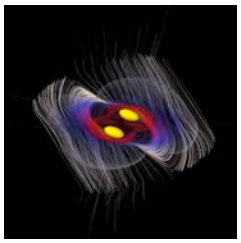


Interactive Monitoring
- Determine job's state and progress
- View current parameters
- List used resources
- How good (or bad) is it performing ?

Computational Steering
- Correct/adjust runtime-steerable parameters
- Modify algorithms
- Enable/disable output
- Kill job

Online Data Visualization
- Analyse the data as they are being calculated

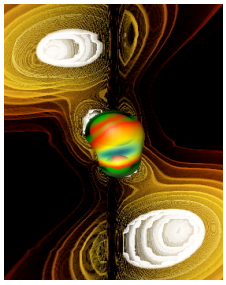
Grand Challenge Collaborations



NASA Neutron Star Grand Challenge
5 US Institutions
Attack colliding neutron star problem

NSF Black Hole Grand Challenge

- 8 US Institutions
- 5 years
- Attack colliding black hole problem



EU Astrophysics Network
10 EU Institutions
3 years
Continue these problems

Examples of Future of Science & Engineering
Require Large Scale Simulations, beyond reach of any machine
Require Large Geo-distributed Cross-Disciplinary Collaborations
Require Grid Technologies, but not yet using them!

Cactus Computational Toolkit

- Cactus is a freely available, modular, portable and manageable environment for collaboratively developing parallel, high-performance multi-dimensional simulations
- Developed originally for Numerical Relativity, but now general framework for parallel computing (CFD, astrophysics, climate modeling, chemical engineering, quantum gravity, …)
- Active user and developer communities, main development now at LSU and AEI.
- Open source, documentation, etc
- http://www.cactuscode.org

Cactus in a Nutshell

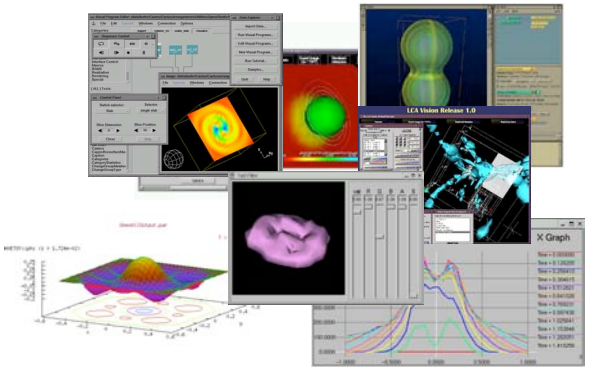
- Cactus acts as the "main" routine, takes care of e.g. parallelism, IO, checkpointing, parameter file parsing, provides different computational infrastructure (reduction operators, interpolators, coordinates, elliptic solvers)
- Everything Cactus "does" is contained in thorns (modules). If you want to use interpolation, you find and add a thorn which does interpolation.
- Extensible, add own interpolators, IO methods etc.
- Provide easy-to-use environment for collaborative, high-performance computing, easy path to Grid computing.

Cactus Output Methods

Most work in 1D/2D or 3D

- Scalars/Reductions to screen
- Scalars/Reductions
- ASCII
- FlexIO, HDF5
- Streaming HDF5
- Panda parallel IO
- Jpegs
- Isosurfaces, Geodesics
- Parallel renderer

Automatic checkpoint and recover (between any machines)



Documentation:

- Overview in thorn CactusBase/IOUtil
- Web page

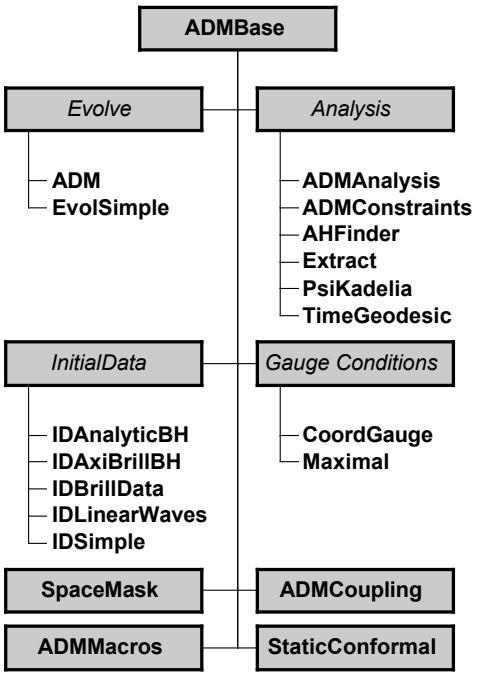
www.cactuscode.org/VizTools

Cactus Einstein Architecture

All thorns agree on some basic (minimal) principles:

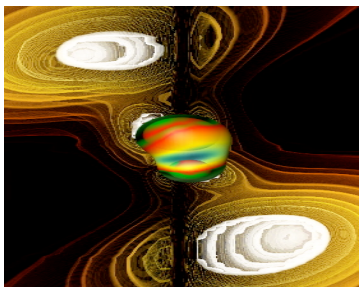
- Names of main variables
- Names of main parameters
- Scheduling of main events

Many community thorns add onto and extend this base



Grid Computing: A New Paradigm

- Computational resources scattered across the world
  - Compute servers (double each 18 months)
  - File servers
  - Networks (double each 9 months)
  - Playstations, cell phones etc…
- How to take advantage of this for scientific simulations?
  - Harness multiple sites and devices
  - Models with new level of complexity and scale, interacting with data



Current Grid App Types

- Community Driven
  - Serving the needs of distributed communities
  - Video Conferencing
  - Virtual Collaborative Environments
    - Code sharing to "experiencing each other" at a distance…
- Data Driven
  - Remote access of huge data, data mining
  - Weather Information systems
  - Particle Physics
- Process/Simulation Driven
  - Demanding Simulations of Science and Engineering
  - Get less attention in the Grid World, yet drive HPC!
- Remote, steered, etc…

New Paradigms for Dynamic Grids

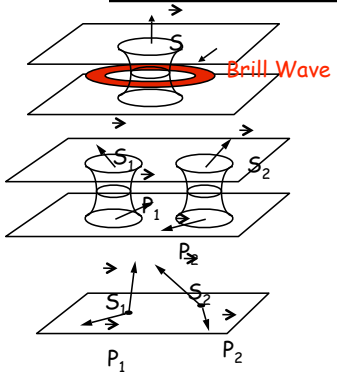
- Addressing large, complex, multidisciplinary problems with collaborative teams of varied researchers …
- Code/User/Infrastructure should be aware of environment
  - Discover resources available NOW, and their current state?
  - What is my allocation on these resources?
  - What is the bandwidth/latency between sites?
- Code/User/Infrastructure should make decisions
  - Slow part of my simulation can run independently … spawn it off!
  - New powerful resources just became available … migrate there!
  - Machine went down … reconfigure and recover!
  - Need more memory (or less!), get by adding (dropping) machines!

New Grid Scenarios

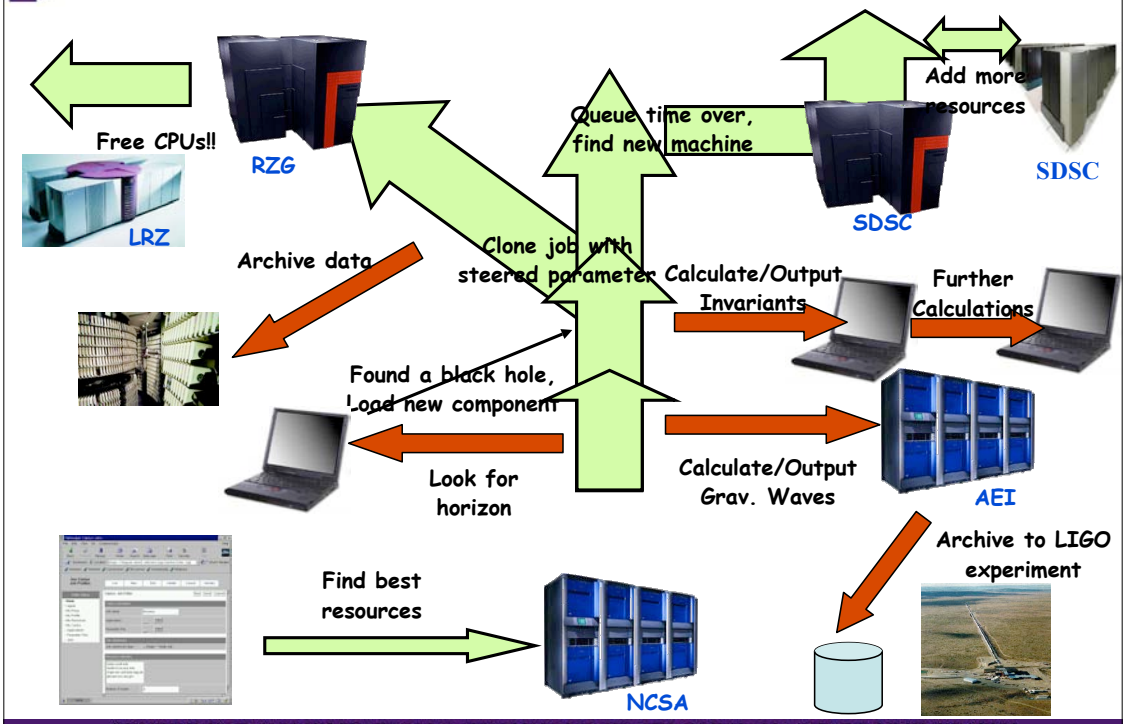
- Intelligent Parameter Surveys, speculative computing
- Dynamic Staging: move to faster/cheaper/bigger machine
- Multiple Universe: create clone to investigate steered parameter
- Automatic Component Loading: needs of process change, discover/load/execute new calc. component on approp.machine
- Automatic Convergence Testing
- Look Ahead: spawn off and run coarser resolution to predict likely future
- Spawn Independent/Asynchronous Tasks: send to cheaper machine, main simulation carries on
- Routine Profiling: best machine/queue, choose resolution parameters based on queue
- Dynamic Load Balancing: inhomogeneous loads, multiple grids

Future Dynamic Grid Computing

Physicist has new idea !



Future Dynamic Grid Computing



GridLab Project

- http://www.gridlab.org
- EU Funded ($6M) by 5th Framework
  - January 2002-March 2005:
- Many partners in Europe and US
  - PSNC (Poland), AEI & ZIB (Germany), VU (Netherlands), MASARYK (Czech), SZTAKI (Hungary), ISUFI (Italy), Cardiff (UK), NTUA (Greece), Chicago, ISI & Wisconsin (US), Sun, Compaq/HP, LSU

GridLab

Information Society Technologies

Grid Application Toolkit

Layer between applications and grid infrastructure:

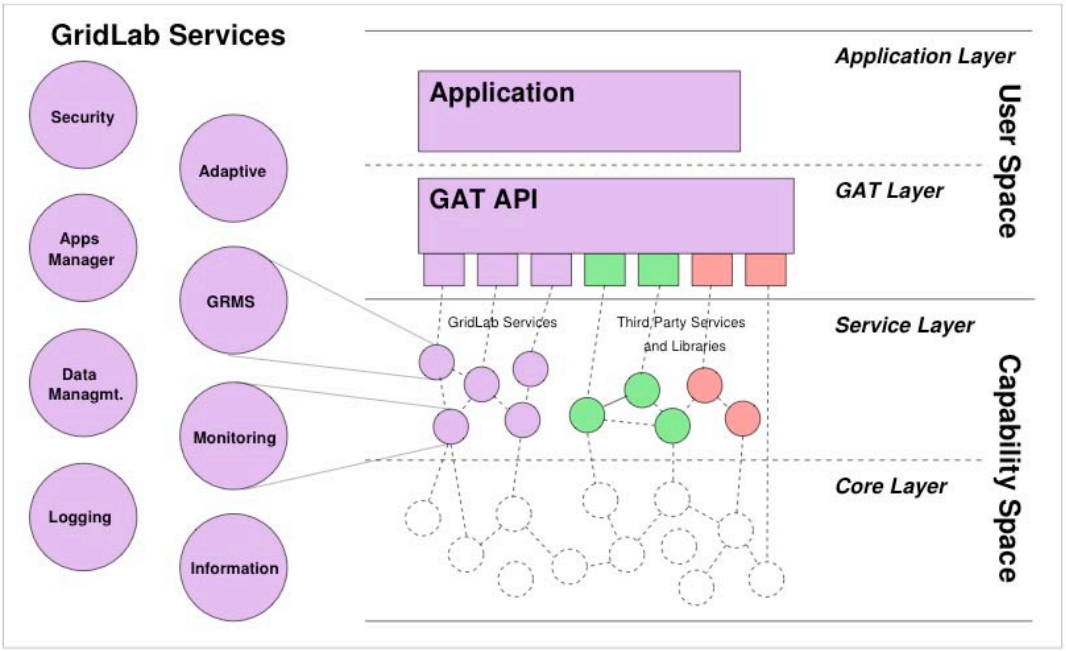
- Higher level than existing grid APIs, hide complexity, abstract grid functionality through application oriented APIs
- Insulate against rapid evolution of grid infrastructure and state of grid deployment
- Choose between different grid infrastructures
- Make it possible for application developers to use and develop for the grid independent of the state of deployment of the grid infrastructure

Motivating influence on SAGA Research Group at GGF

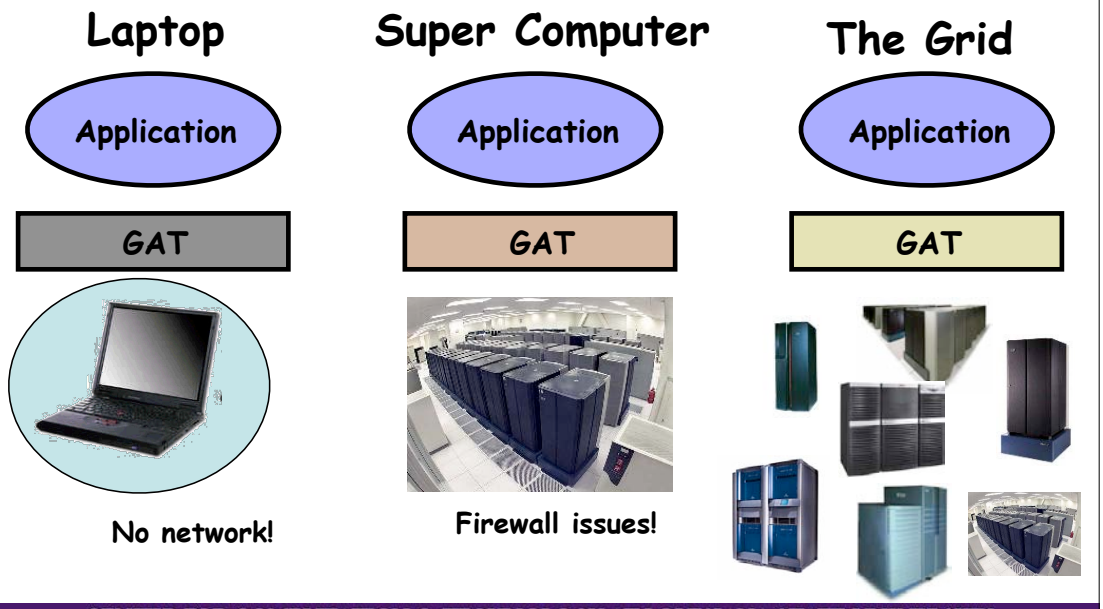
GAT: Grid Application Toolkit

- Standard API and Toolkit for developing portable Grid applications independently of the underlying Grid infrastructure and available services
- Implements the GAT-API
  - Used by applications
- GAT Adaptors
  - Connect to capabilities/services
- GAT Engine
  - Provides the function bindings for the GAT-API
- http://www.gridlab.org/GAT

GAT Architecture



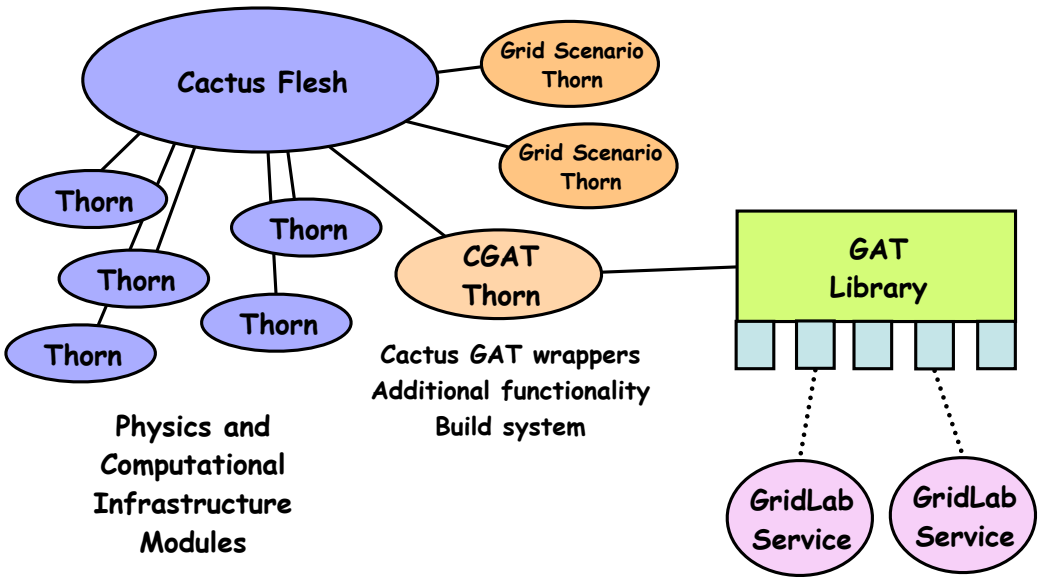
The Same Application …



GAT Distribution

- GAT Engine releases at http://www.gridlab.org/WorkPackages/wp-1/gatreleases.html
  - C, C++, Python, Java, .NET bindings
  - Examples, and "one liner" client tools
- GAT Adaptors
  - Default Adaptors:
    - Basic functionality which will work on a single isolated machine (e.g. cp, fork/exec)
  - GridLab Adaptors:
    - Connect to services developed through the GridLab project (GRMS, Mercury, Delphoi, iGrid)
  - Globus Adaptors (LSU):
    - Core Globus functionality: GRAM, MDS, GT-RLS, GridFTP
  - Others: DRMAA, Condor, SGE, gsissh, SRB, Curl, …

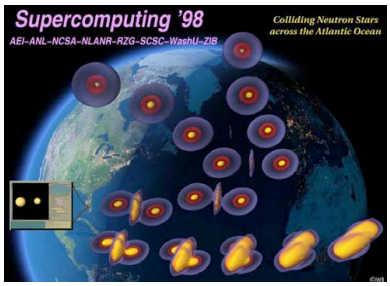
Cactus/GAT Integration



Distributed Computation

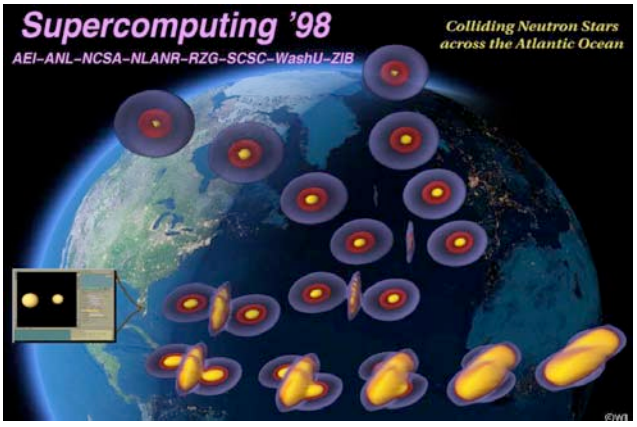
Harnessing Multiple Computers …

- Why would anyone want to do this?
  - Capacity: can run larger models if can use more machines
  - Throughput: can run models sooner
- Issues
  - Wide Area Network between machines means that optimizing communications between processors is crucial
  - Techniques, such as data compression, have been developed and tested for this



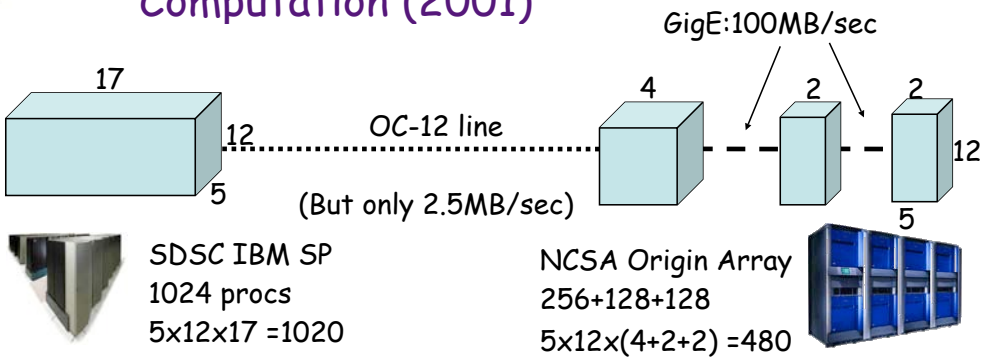
Metacomputing in '98

- SC93 - SC2000
- Typical scenario
  - Find remote resource (often using multiple computers)
  - Launch job (usually static, tightly coupled)
  - Visualize results (usually in-line, fixed)
- Need to go far beyond this
  - Make it much, much easier
    - Portals, Globus, standards
  - Make it much more dynamic, adaptive, fault tolerant
  - Migrate this technology to general user



Metacomputing Einstein's Equations: Connecting T3E's in Berlin, Garching, SDSC

Dynamic Adaptive Distributed Computation (2001)



These experiments:

- Einstein Equations (but could be any Cactus application)

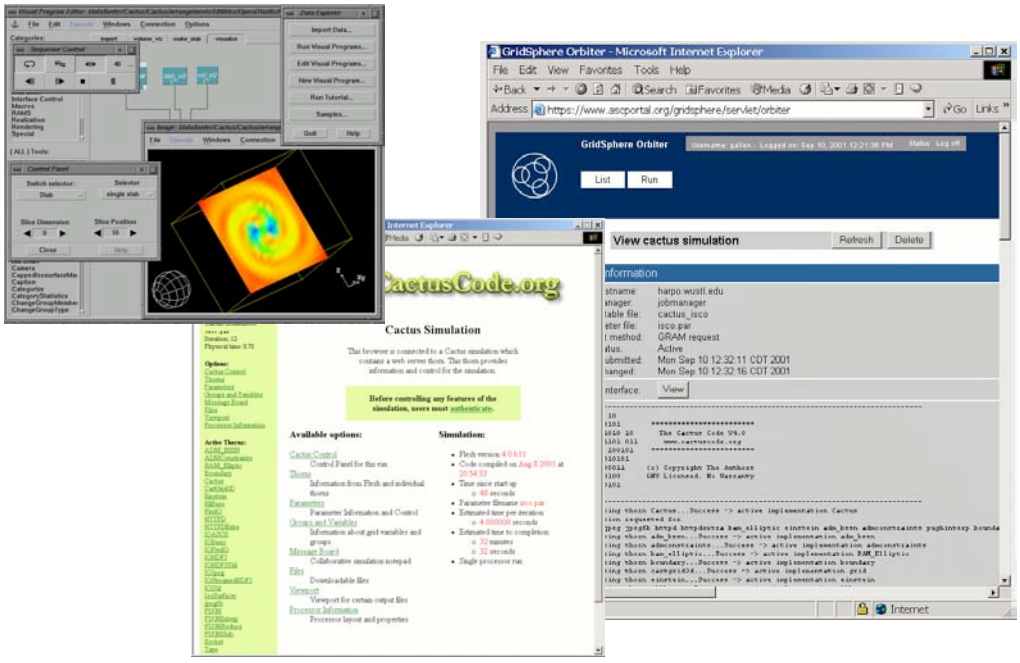
Achieved:

- First runs: 15% scaling
- With new techniques: 70-85% scaling, ~ 250GF

Dynamic Adaptation: Number of ghostzones, compression, …

Won "Gordon Bell Prize" (Supercomputing 2001, Denver)

Start simple: sit here, compute there … Astrophysics Simulation Collaboratory (2001)

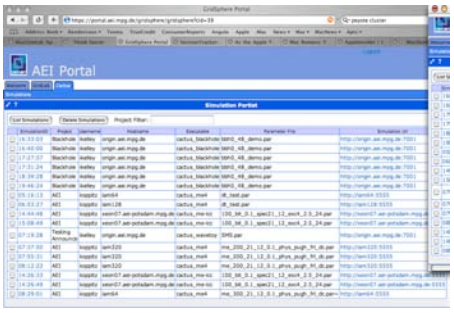


Remote Monitoring/Steering: Thorn HTTPD

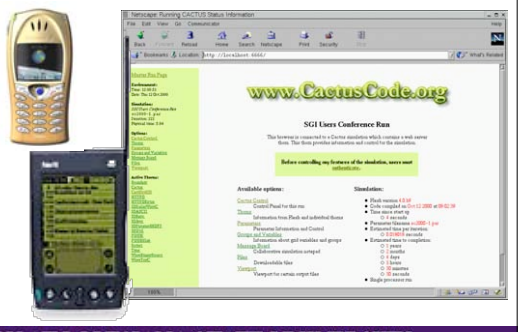
- Thorn which allows any simulation to act as its own web server
- Connect to simulation from any browser anywhere … **collaborate**
- Monitor run: parameters, basic visualization, …
- Change steerable parameters
- See running example at www.CactusCode.org
- Wireless remote viz, monitoring and steering

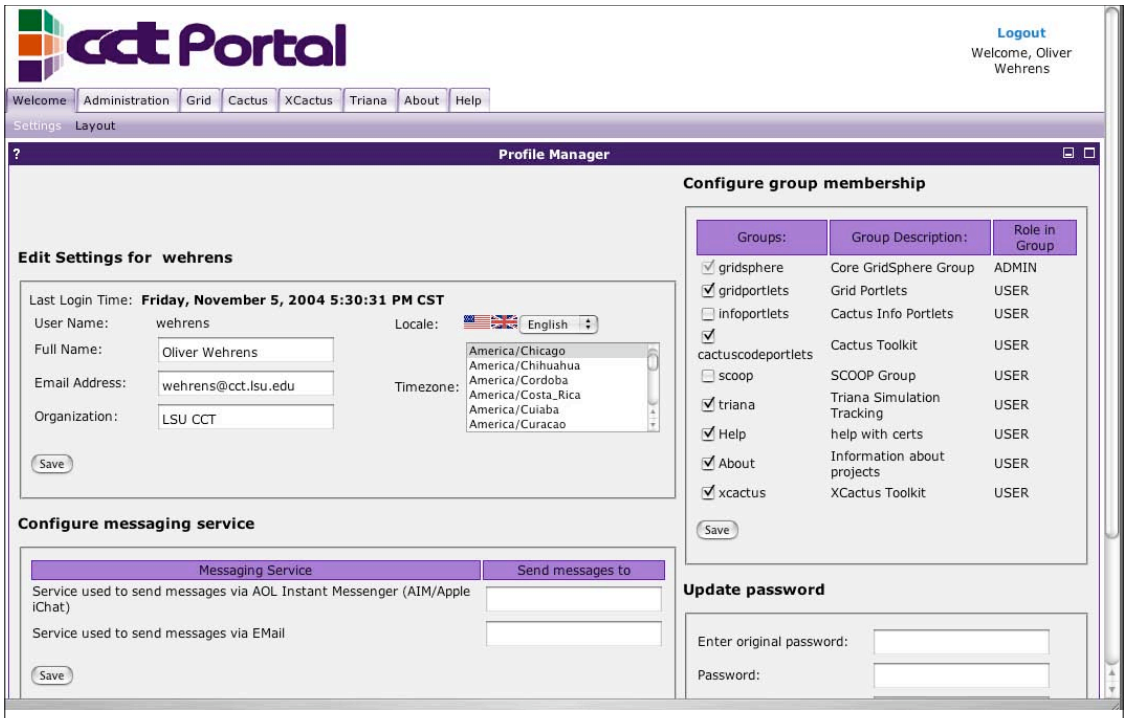
Cactus Portal with GridSphere Framework (www.gridsphere.org)

- Coordinating point for group and collaborators
- Simulation staging, tracking, steering, archiving
- Introduction of new (Grid) technologies
- New portlets easily added

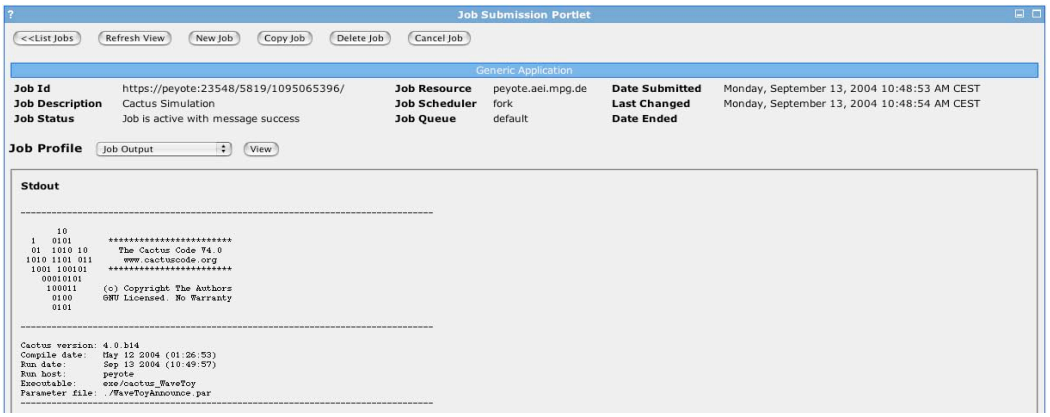


GridSphere:
New implementation of portlet API
Simple and effective framework for Grid & collaborative portals
GPDK/ASC portal authors
Grid portlets: security, start jobs, move files, notification, monitoring

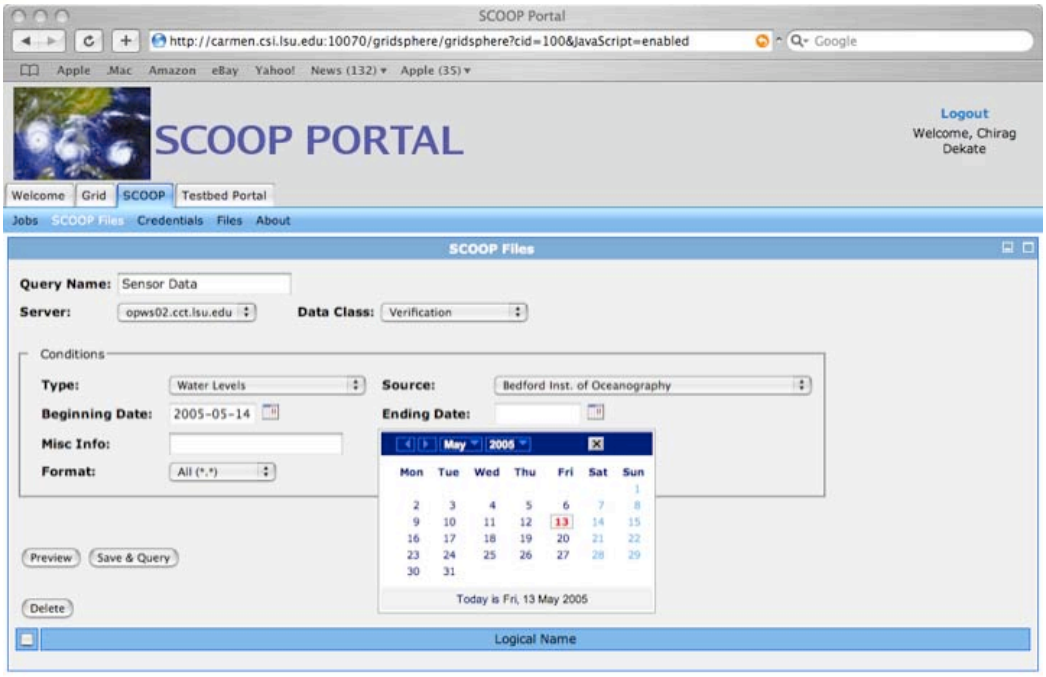




Job History & Output

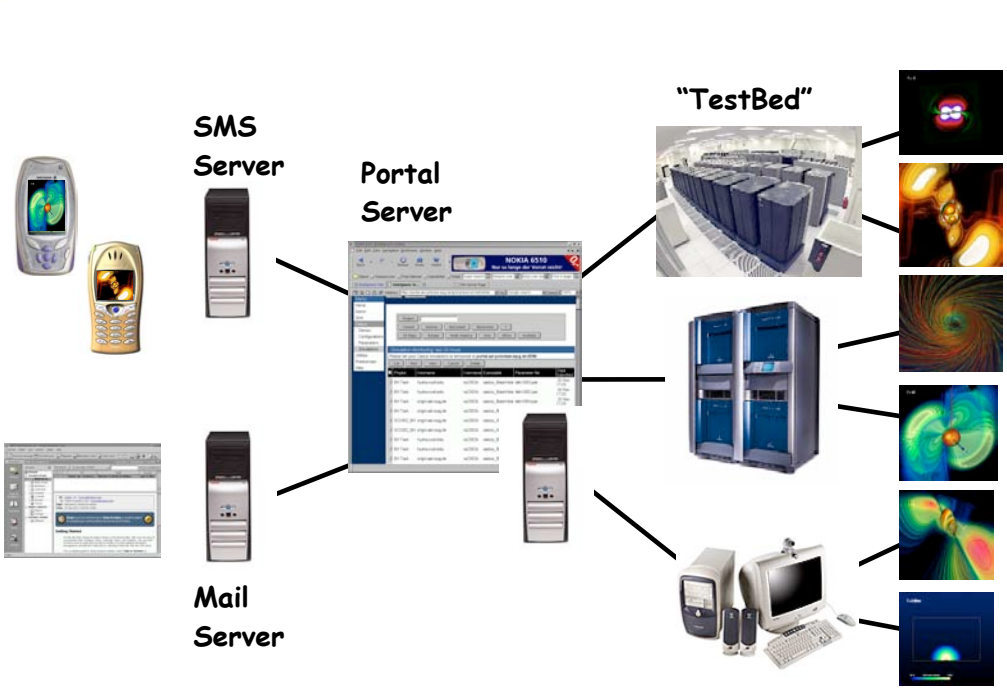


Logical File Browser

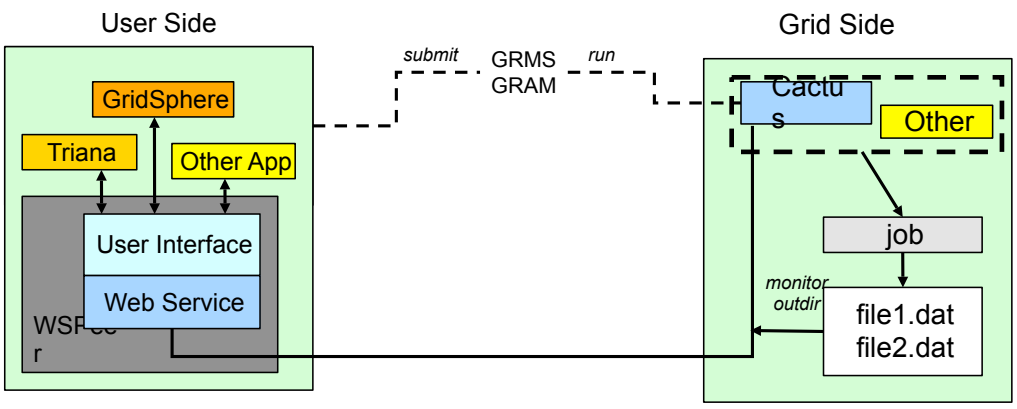


May 13, 2005

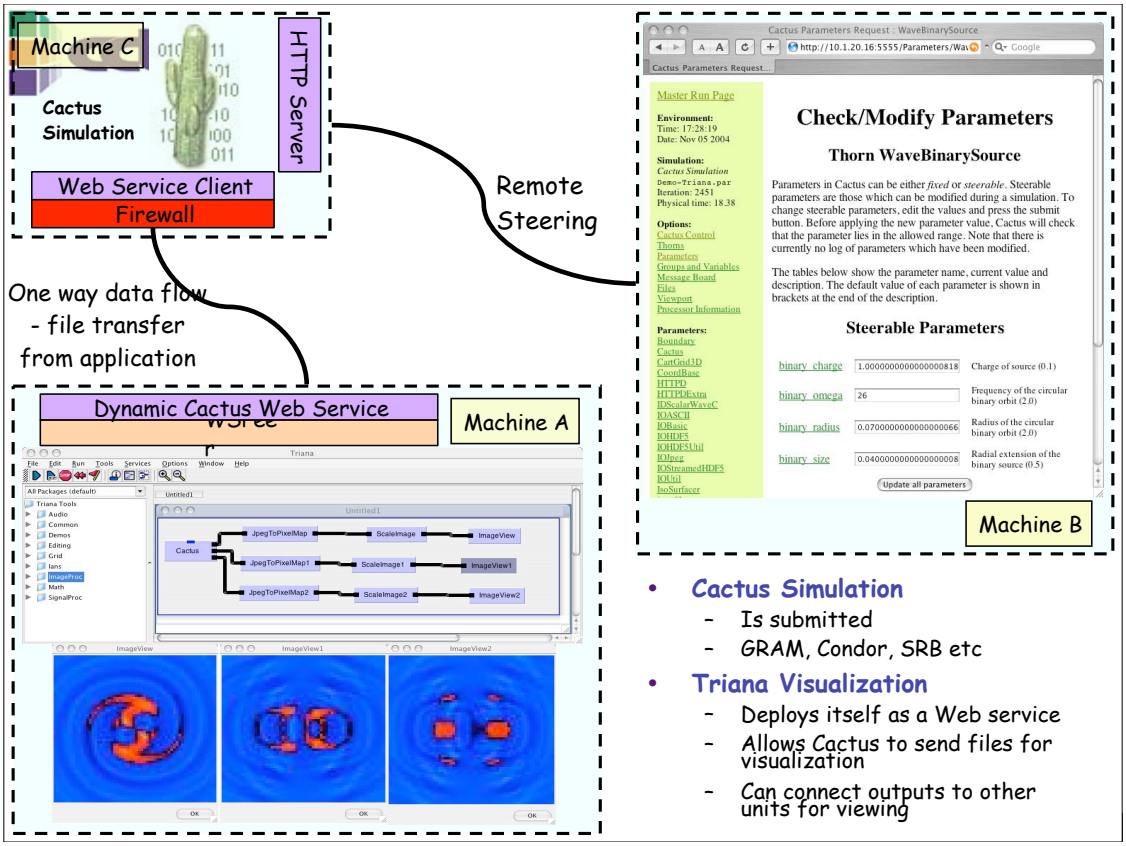
Notification



Generic Monitoring Framework

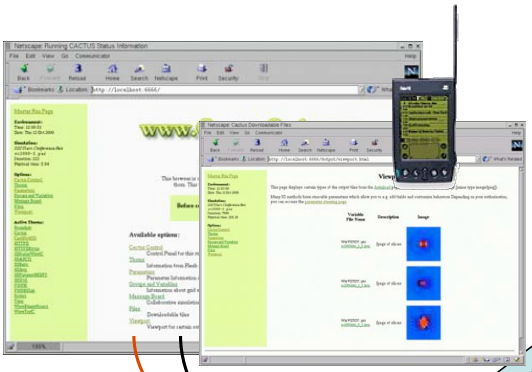


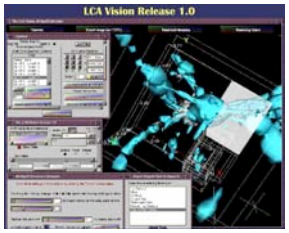
Invoke using SOAP



- **Cactus Simulation**
  - Is submitted
  - GRAM, Condor, SRB etc
- **Triana Visualization**
  - Deploys itself as a Web service
  - Allows Cactus to send files for visualization
  - Can connect outputs to other units for viewing

Remote Visualization & Steering





Any Viz Client:
LCA Vision, OpenDX
Changing any steerable parameter
- Parameters
- Physics, algorithms
- Performance

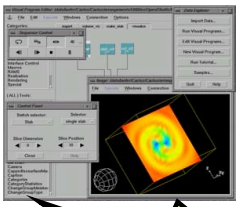
Remote Viz data

Streaming HDF5
Autodownsample

Remote Viz data

Remote File Access

SC02 (Baltimore)



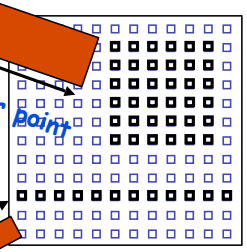
Visualization

HDF5 VFD/
GridFTP:
Clients use file URL
(downsampling,hyperslabbing)

Lapse for r<1, every other point

More Bandwidth Available

Network Monitoring Service



TB Data File

Remote Data Storage

NCSA (USA)

grr,psi on timestep 2

AEI (Germany)

Analysis of Simulation Data

Cactus Worm: SC2000

- Cactus simulation starts, launched from a portal
- Migrates itself to another site
- Registers new location
- User tracks/steers, using http, streaming data, etc...

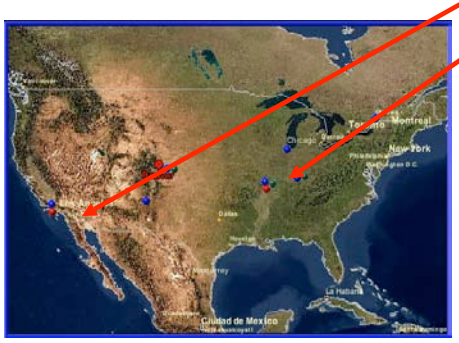
- Continues around Europe..

Spawning: SC2001 Demo

- Black hole collision simulation
  - Every n timesteps, time consuming analysis tasks done
  - Process output data, find gravitational waves, horizons
  - Can take much time
  - Processes do not run well in parallel
- Solution: Use "Spawner" thorn with Cactus
- Analysis tasks outsourced
  - Resource Discovery
  - Login, data transfer
  - Remote jobs started up
- Main simulation can keep going without pausing
  - Except to spawn: may be time consuming itself
- It worked!

SC2001: Spawning on ARG Testbed

Main Cactus BH
Simulation starts here



All analysis tasks spawned automatically to free resources worldwide

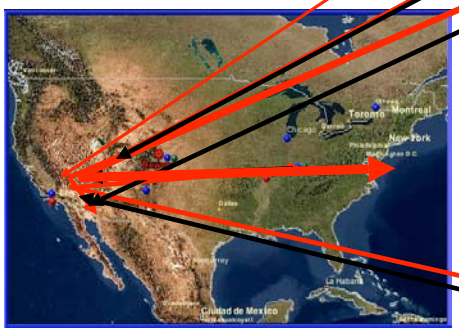
User only has to invoke Cactus "Spawner" thorn…

Task Farming/Steering Combo

- Large complex simulation, dozens of input parameters
  - Selection is trail and error (resolution, boundary, etc)
- Remember look ahead scenario? Run at lower resolution predict likely outcome!
  - Task farm dozens of smaller jobs across grid to test initial parameters for big run
  - Task farm manager sends out jobs to resources, collects results
  - Lowest error parameter set chosen
- Main simulation steered using "best" parameters

SC2002 Demo

Main Cactus BH Simulation started in California



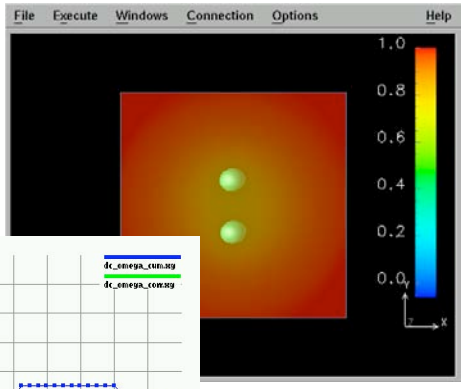
Dozens of low resolution jobs sent out to test parameters

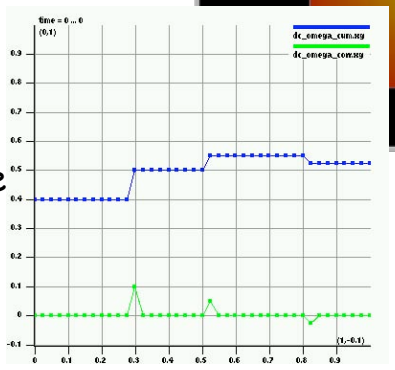
Data returned steers main job

Huge job generates remote data to be visualized in Baltimore

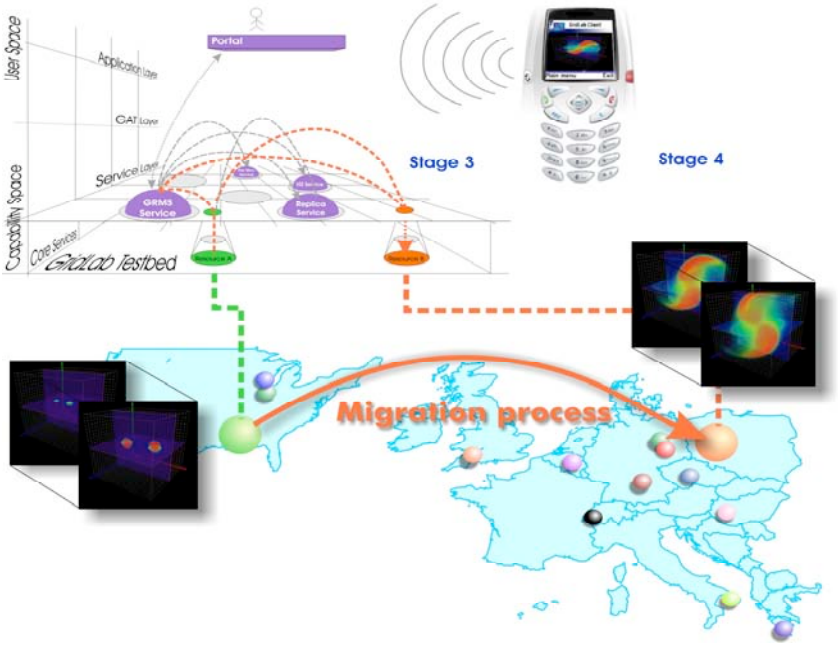
Grid-Black Holes

- Task farm small Cactus black hole simulations across testbed
- Parameter survey: black hole corotation parameter
- Results steer a large production black hole simulation



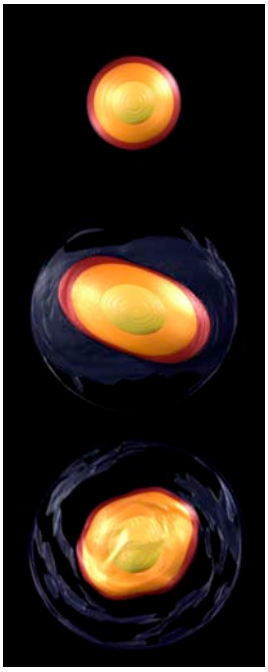


Migration (Cactus Worm revisited)



Migration of Legacy Applications

- Working with legacy astrophysics code:
  - 3D neutron star code (Fortran, MPI)
    - Shangli Ou, Joel E. Tohline (LSU), Lee Lindblom (Caltech)
  - Code models the non-linear development of the secular bar-mode instability that is driven by gravitational radiation-reaction (GRR) forces in rotating neutron stars!!!
  - Important for providing templates for gravitational wave detectors (LIGO)
- Use a GAT wrapper to migrate application between resources



Conclusion

- Many prototypes for how Grid technologies can be used to good effect to enable old and new scenarios in astrophysics and other application areas.
- Now using Grid Application Toolkit to make these scenarios robust, common place, and infrastructure/ service independent for wide production use
- Application software itself is very important to be able to really use Grids
- Links:
  - http://www.cactuscode.org
  - http://www.gridlab.org/GAT
  - http://www.gridsphere.org
  - http://wiki.cct.lsu.edu/saga

Grid Introduction

Gabrielle Allen

Center for Computation & Technology

Louisiana State University

gallen@cct.lsu.edu

May 17, 2005

Center for Computation & Technology

What is Grid Computing?

Look to Googlism.com …

- ☑ grid computing is an explosive moment in the development of computing brainpower
- ☑ grid computing is not "years" away but rather here today
- ☑ grid computing is really "the next big thing"
- ☑ grid computing is going to enable things we can't even imagine
- ☑ grid computing is distributed computing
- ☑ grid computing is no longer just a buzzword
- ☑ grid computing is like throwing darts at a dragonfly
- ☑ grid computing is a step up from distributed computing
- ☑ grid computing is one of the hottest areas in computing today
- ☑ grid computing is a flaky concept

Learning Grid Computing

- Not many "textbooks" on Grid Computing!!!
  - Good general overview …
    - **The Grid 2: Blueprint for a New Computing Infrastructure.** Ed:Ian Foster and Carl Kesselmann
  - More technical …
    - **Grid Computing, Making the Global Infrastructure a Reality.** Ed:Berman, Fox, Hey
  - Search for Grid Computing courses at universities
- It is a rapidly moving research area
  - Many projects/research based on Globus V.2 but Globus V.4 already released!
- Best way to learn is by trying it out yourself
  - most software is free, just one machine can be a Grid

APIs and Protocols are Important

- Protocols: agreed format for transmitting data between two devices
- API: specification for a set of routines to facilitate application development
- SDK: programming package that enables a programmer to develop applications for a specific platform, instantiation of an API
- Standard APIs/SDKs are important
  - Enable application portability, but w/o standard protocols, interoperability hard (every SDK speaks every protocol?)
- Standard protocols are important
  - Enable cross-site interoperability
  - Enable shared infrastructure
  - But w/o standard APIs/SDKs, application portability is hard (different platforms access protocols in different ways)

What is a Grid?

- 1965, Fernando Corbato:

  computer facility "like a power company or water company"
- 1969, Len Kleinrock:

  "We will probably see the spread of 'computer utilities', which, like present electric and telephone utilities, will service individual homes and offices across the country."
- 1998, Kesselman & Foster:

  "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."
- 2000, Kesselman, Foster, Tuecke:

  "…coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations."

I-WAY: SC95

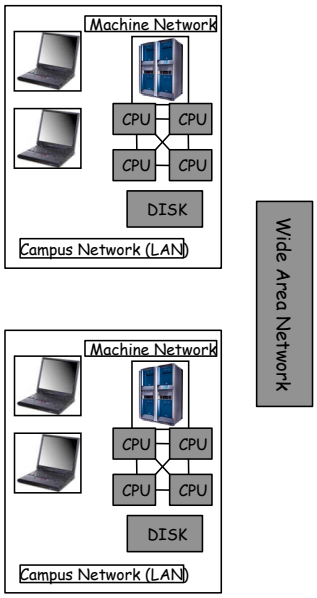
- High speed experimental distributed computing project.
- Set up ATM network connecting supercomputers, mass storage, advanced viz devices at 17 US sites.
- 30 software engineers, 60 applications, 10 networks (most OC-3c/155Mbps)
- Application focused (remote viz, metacomputing, collaboration)
- Single interface to schedule and start runs
- I-POP machines (17) coordinated I-WAY "virtual machines", gateways to the I-WAY
- I-Soft software for management/ programming

Ian Foster's Grid Checklist (2002)

- A Grid is a system that:
  - Coordinates resources that are not subject to centralized control
  - Uses standard, open, general-purpose protocols and interfaces
  - Delivers non-trivial qualities of service

Components for Grid Computing

- Distributed People
  - Research communities who need to share data, or codes, or computers, or equipment to work on and understand common problems
  - Astrophysics Network: relativists, astrophysicists, computer scientists, mathematicians, experimentalists, data analysts
- Distributed Resources
  - Computers: supercomputers, clusters, workstations, PDAs
  - Storage devices, databases,



Components for Grid Computing

- Software infrastructure
  - Links all these together
  - Low level: security, information, communication, …
  - Middleware: data management, resource brokers, portlets, monitoring, workflow, …
- Examples
  - Globus (low level)
  - Condor (higher level)

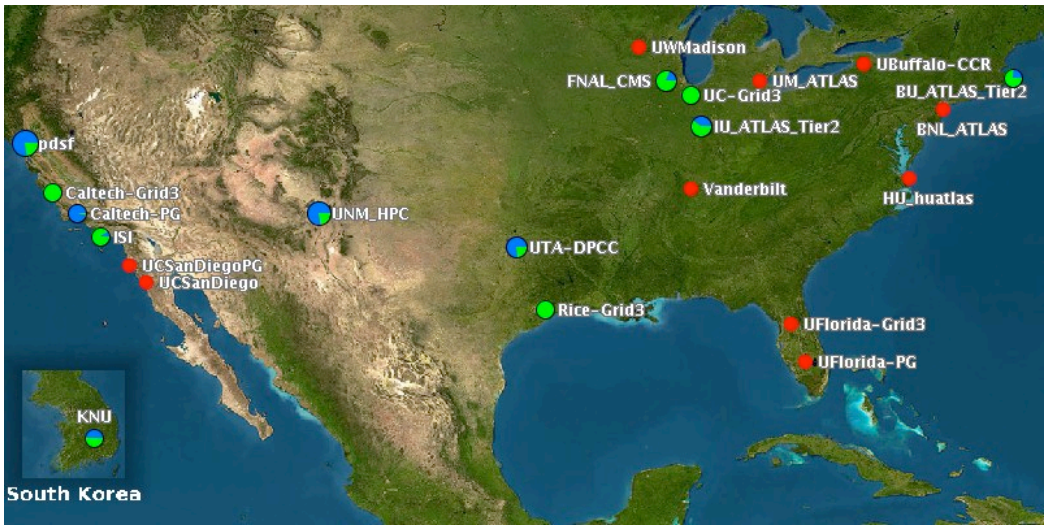
Virtual Organizations

- Groups of organizations that use the Grid to share resources for specific purposes
  - EU DataGrid, Alliance, TeraGrid, SC02 Global Grid Testbed, etc
- Support a single community or multiple communities
- Deploy compatible technology and agree on working policies
- Deploy different network accessible services:
  - Grid Information
  - Grid Resource Brokering
  - Grid Monitoring
  - Grid Accounting
- Authentication and Authorization

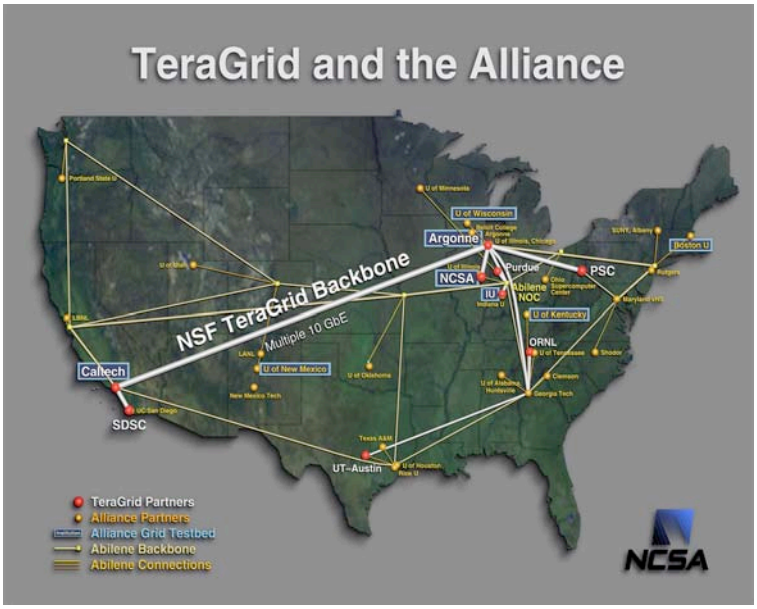
Components for Grid Computing

- The Applications !!!!
- Application Level Utilities
  - Standard toolkits
  - SDKs
  - Libraries
  - User portals
- Applications themselves
  - What properties do they need to run on a Grid?
  - Need to be highly portable and machine independent
- Development tools (debuggers, profilers, …)

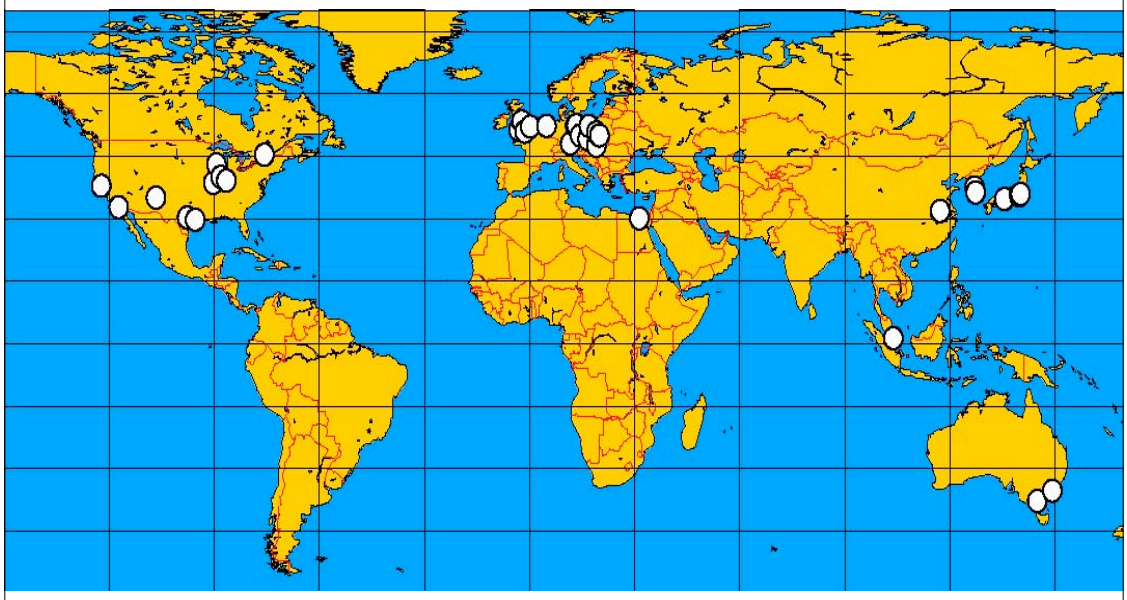
The Grids: Grid2003



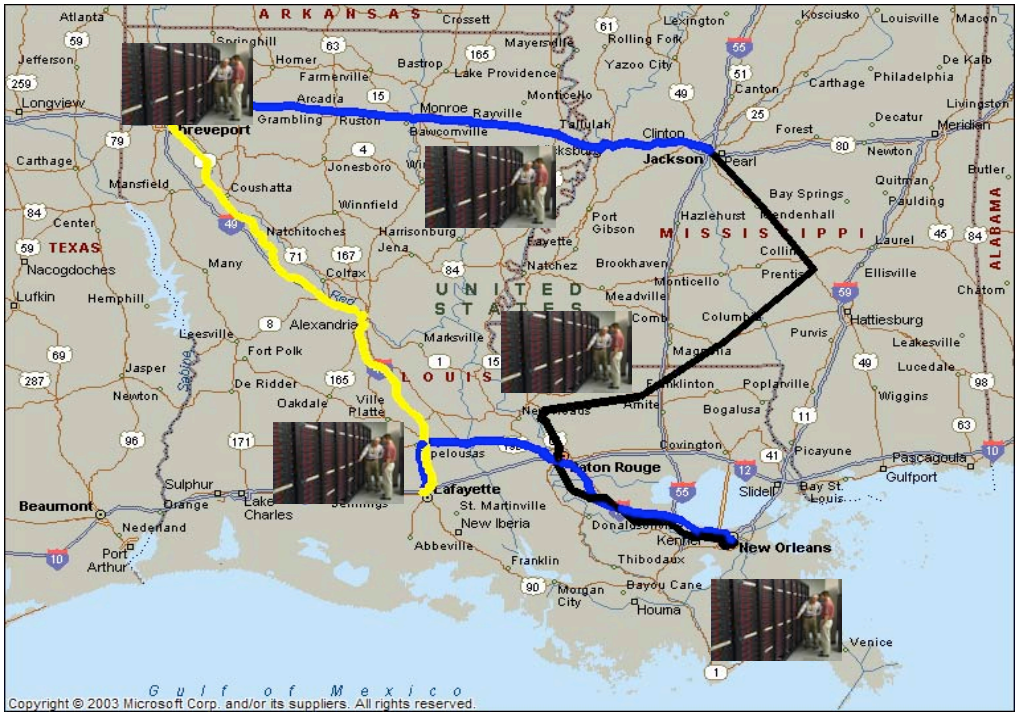
The Grids: TeraGrid



The Grids: Global Grid Testbed Collaboration



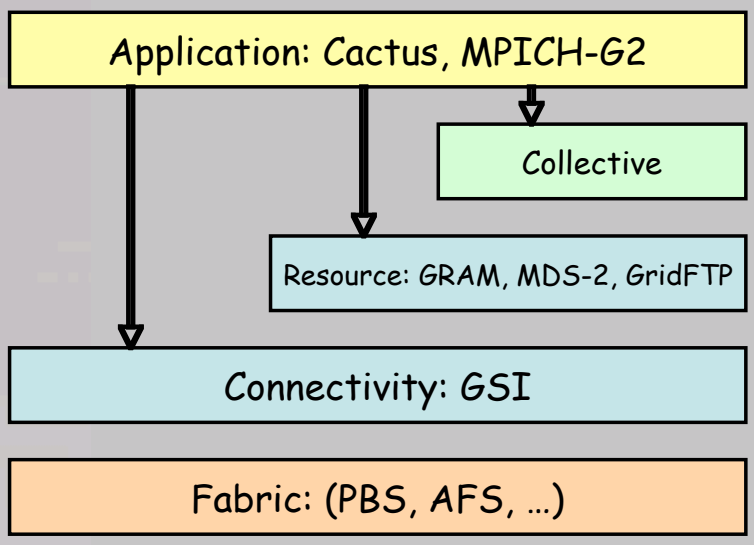
The Grids: LONI



Basic Grid Infrastructure:

- Globus Toolkit:
  - Security infrastructure: GSI
  - Running jobs: GRAM/Condor-G
  - Transferring data: GridFTP
  - Discovering information: MDS
- Others: Unicore, seti@home, Condor, iGrid, …

Grid Architecture: GT2



Grid Services

- Monitoring resources and applications
- Security (Authentication and Authorization)
- Data catalogues and replication
- Notification of events
- Resource brokering
- Information providers
- Application managers
- Logging
- Application services

Why Grid Security is Hard

- Resources may be valuable & the problems being solved sensitive
- Resources are often located in distinct administrative domains
  - Each resource has own policies, procedures, security mechanisms, etc.
- Implementation must be broadly available & applicable
  - Standard, well-tested, well-understood protocols; integrated with wide variety of tools

Security: Terminology

- Authentication: Establishing identity
  - e.g. passport, driving license, public keys
- Authorization: Establishing rights
  - what are you allowed to do
- Message protection
  - Message integrity: make sure noone tampers with it
  - Message confidentiality: make sure noone else can read it
- Non-repudiation: guarantee message sent and received
- Digital signature: guarantee sender and contents
- Accounting
- Delegation

Grid Security Infrastructure (GSI)

- Users:
  - Easy to use
  - Single sign-on: only type your password once
  - Delegate proxies
- Administrators
  - Can specify local access controls
  - Have accounting

GSI: How Do We Get These Features?

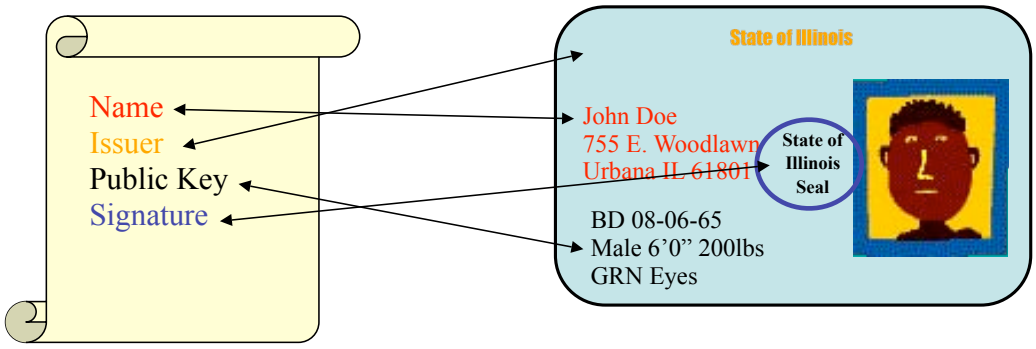
- From the Public Key Infrastructure: PKI
- PKI allows you to know that a given key belongs to a given user
- PKI builds off of asymmetric encryption:
  - Each entity has two keys: public and private
  - Data encrypted with one key can only be decrypted with other
  - The public key is public
  - The private key is known only to the entity
- The public key is given to the world encapsulated in a X.509 certificate

Grid Certificates

- On the Grid, each user and service is identified via a GSI certificate, which includes
  - A subject name, which identifies the person or object that the certificate represents.
  - The public key belonging to the subject.
  - The identity of a Certificate Authority (CA) that has signed the certificate to certify that the public key and the identity both belong to the subject.
  - The digital signature of the named CA
- GSI certificates are encoded in the X.509 certificate format.

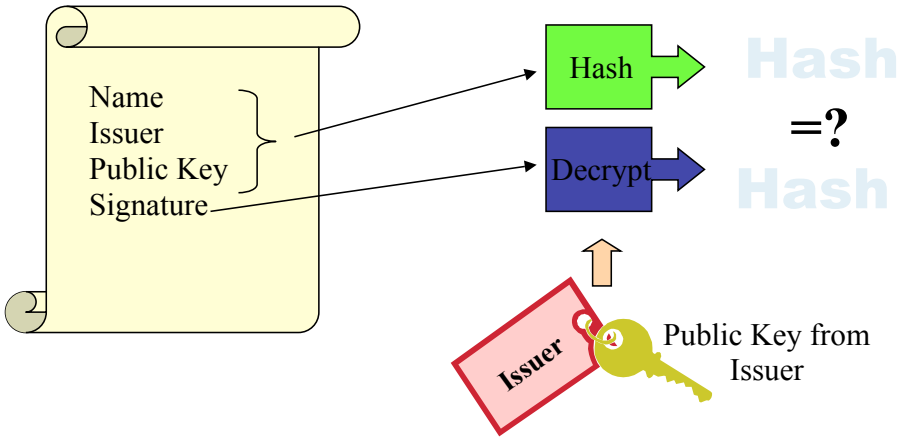
GSI: What is a Certificate?

- Similar to passport or driver's license: Identity signed by a trusted party

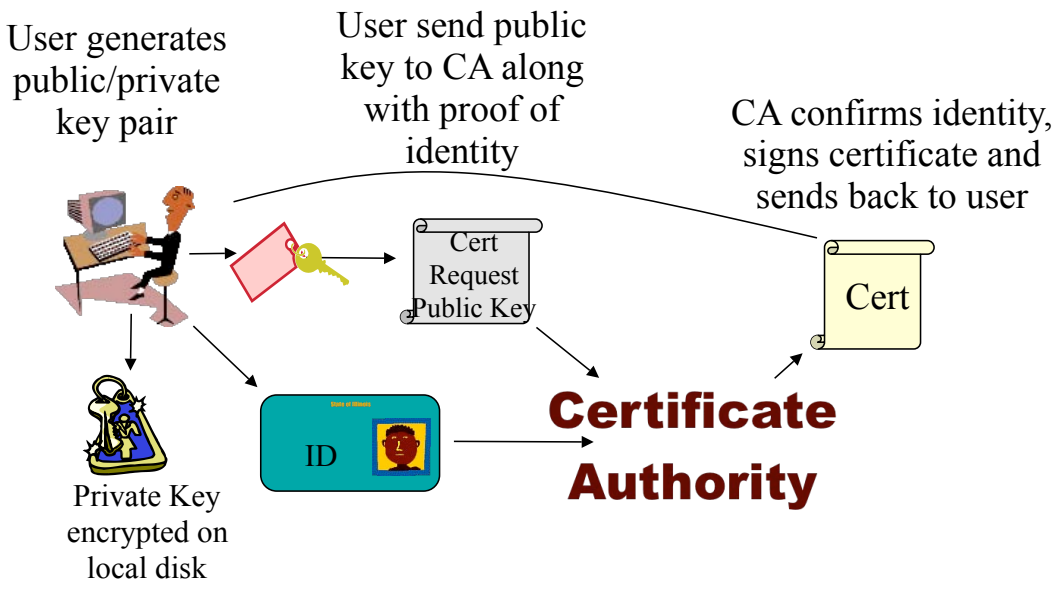


GSI: Certificates

- By checking the signature, one can determine that a public key belongs to a given user



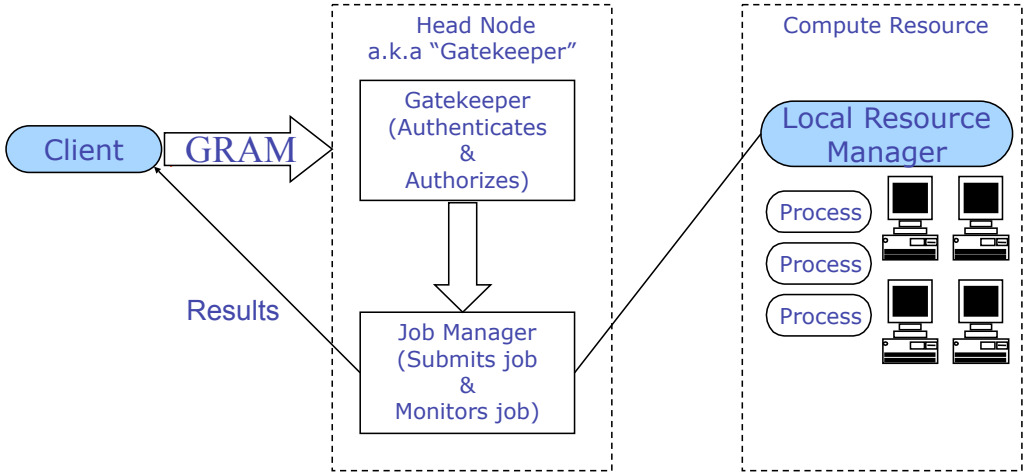
How Do You Get a Certificate?



GRAM: What is it?

- Given a job specification:
  - Create an environment for a job
  - Stage files to/from the environment
  - Submit a job to a local scheduler
  - Monitor a job
  - Send job state change notifications
  - Stream a job's stdout/err during execution

GRAM: How Does it Work?



GRAM: What is a "Local Resource Manager?"

- It's usually a batch system that allows you to run jobs across a cluster of computers
- Examples:
  - Condor
  - PBS
  - LSF
  - Sun Grid Engine
  - (fork … runs a process on the gatekeeper)

GRAM: RSL

- The client describes the job with the Resource Specification Language (RSL)

```
& (executable = a.out)
  (directory = /home/nobody )
  (arguments = arg1 "arg 2")
```

- It is most convenient to use higher level tools (such as portals) to construct anything but simple RSL
- http://www.globus.org/gram/rsl_spec1.html

GRAM: Security

- GRAM uses GSI for security
- Submitting a job requires a full proxy
  - The remote system & your job will get a limited proxy
  - The job will run—you had a full proxy when you submitted
  - But your job cannot submit other jobs

GridFTP: What is it?

- A secure, robust, fast, efficient, standards based, widely accepted data transfer protocol
- An implementation:
  - Globus provides a server
  - Globus provides a client: globus-url-copy
  - Other people provide clients: uberftp

GridFTP: Features

- Security through GSI
  - Note that GSI can provide encryption in addition to authentication and authorization
- Reliability by restarting failed transfers
- Fast
  - Can set TCP buffers for optimal performance
  - Parallel transfers
  - Striping (multiple endpoints)
- Not all features easily accessible from basic client

GridFTP: Basic Use

- globus-url-copy file:fullpath/file gsiftp://host/path/file
  - File: URL refers to a local file
  - Gsiftp: URL refers to a remote file, accessed with GridFTP
- You can specify two gsiftp URLs to do third-party transfers
- You can specify other URLs, including http & https

Globus Monitoring and Discovery Service

- MDS is a grid information service
- It provides:
  - Uniform, flexible access to information
  - Scalable, efficient access to dynamic data
  - Access to multiple information sources
  - Decentralized maintenance
- Based on LDAP

Information for What?

- Operation of Grid
  - Monitoring and testing Grid
- Deployment of applications
  - What resources are available to me? (Resource discovery)
  - What is the state of the grid? (Resource selection)
  - How to optimize resource use? (Application configuration and adaptation)
- Information for other Grid Services to use

What are the Problems

- How to obtain needed information? (automatic and accurate)
- Information is always old
  - Resources change state
  - Takes time to retrieve information
  - Need to provide quality metrics
- Grid is distributed
  - global state is very complex
  - Scalability, efficiency and overhead
- Component failure
- Security

Grid Information

- Compute Resource Specific
  - Name of resource, IP address, site name, location, firewalls, names of administrators, scheduled downtimes
  - Machine type (SMP, ccNUMA, number of processors, interconnects)
  - Processor types and characteristics (vendor, OS, cache, clockspeed)
  - Software installations (software version, location, license)
  - Jobs (queue names and properties, current running and queued jobs)

Grid Information

- Network Specific
  - Network type (peak speed, physical characteristics)
  - Network properties (bandwidth, jitter, latency, QoS)
  - Scheduled downtimes
- Storage Resource Specific
  - File system locations
  - File system properties
  - Current space

Needed Capabilities

- Queryable across a network
- Supports virtual organizations
- Complex queries (search for all linux machines with at least 1GB memory and MPI-LAM installed)
- Authentication and authorization.
- Multiple information providers
- Extensible information schemas
- Efficient return of information
- Extensible to large numbers of resources
- Up-to-date information!!
- Queryable in multiple ways (clients, web, APIs)

Globus MDS

- Monitoring and Discovery Service
  - Set of information service components for publishing and discovering information
  - Single standard interface and scheme to information services in a virtual organization
- MDS can aggregate information from multiple sites each with multiple resources
- Information about each resource is provided by a information provider
- Handles static and dynamic information
- Access restricted by GSI credentials and authorization

MDS Components

- LDAP 3.0 Protocol Engine
  - Based on OpenLDAP with custom backend
  - Integrated caching
- Information providers
  - Delivers resource information to backend
- APIs for accessing & updating MDS contents
  - C, Java, PERL (LDAP API, JNDI)
- Various tools for manipulating MDS contents
  - Command line tools, Shell scripts & GUIs

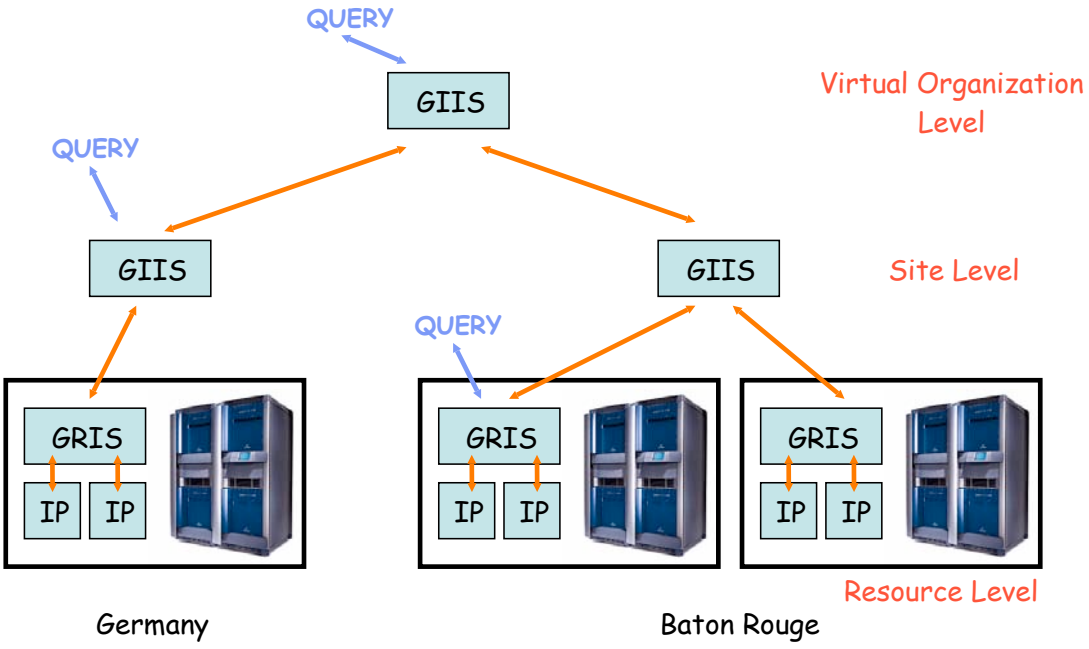
Local Resource Monitoring

- Publishes information to MDS
  - Cluster monitoring: e.g. Ganglia
  - Queue information: GRAM Reporter
  - Network information: NWS
  - Other local monitoring systems may require writing MDS interfaces

Two Classes Of MDS Servers

- Grid Resource Information Service (GRIS)
  - Supplies information about a specific resource
  - Configurable to support multiple information providers
  - LDAP as inquiry protocol
- Grid Index Information Service (GIIS)
  - Supplies collection of information which was gathered from multiple GRIS servers
  - Supports efficient queries against information which is spread across multiple GRIS server
  - LDAP as inquiry protocol

MDS Hierachy



Grid Toolkits

- Basic Grid infrastructure provides core functionality such as remote job submission, data transfer, information, security
- Globus 4.0 provides web service interfaces
- Much more than this is needed for Grids to be useful for real applications
  - Higher level services providing Grid scheduling, accounting, data management, etc.
  - Grid Toolkits making services usable for applications

GAT Motivation

- Grids and Grid middleware are everywhere
- Grid applications are lagging behind, big jump from prototypes and demonstrations to real production use of Grids.
- Problems:
  - Missing or immature grid services
  - Changing environment
  - Different and evolving interfaces to the "grid"
  - Interfaces are not simple for scientific application developers

- Application developers accept Grid computing paradigm only slowly

Simplicity!

- The key objective for application programmers (i.e. a physicist, chemist, engineer)
- Simple API's should be:
  - Be easy to use
    - Simple, finite, consistent API which allows error tracing
  - Be invariant: make upgrades really, really simple
    - Well defined API which rarely changes.
    - Implementation which allows dynamic exchange of key elements and provides runtime abstractions
  - Avoid refactoring/recoding/recompilation
    - Same applications runs today and tomorrow; here and there; on Globus and Unicore; on Globus 2.2.4 and Globus 2.4; on Linux and on Mac; local and on grid;
  - Focus on well-known programming paradigms
    - (e.g., for a file provide a file API – without services to services to files. . .)
      Files are best example: expect open, close, read, write, seek. Do not introduce fancy things like the need to ask a service discovery service to tell me the location of an service which is able to tell me the location of my file..

Copy a File: GASS

```
int RemoteFile::GetFile (char const* source,
                         char const* target) {
globus_url_t                    source_url;
globus_io_handle_t              dest_io_handle;
globus_ftp_client_operationattr_t  source_ftp_attr;
globus_result_t                 result;
globus_gass_transfer_requestattr_t source_gass_attr;
globus_gass_copy_attr_t         source_gass_copy_attr;
globus_gass_copy_handle_t       gass_copy_handle;
globus_gass_copy_handleattr_t   gass_copy_handleattr;
globus_ftp_client_handleattr_t  ftp_handleattr;
globus_io_attr_t                io_attr;
int                             output_file = -1;

if ( globus_url_parse (source_URL, &source_url) != GLOBUS_SUCCESS ) {
  printf ("can not parse source_URL \"%s\"\n", source_URL);
  return (-1);
}

if ( source_url.scheme_type != GLOBUS_URL_SCHEME_GSIFTP &&
     source_url.scheme_type != GLOBUS_URL_SCHEME_FTP    &&
     source_url.scheme_type != GLOBUS_URL_SCHEME_HTTP   &&
     source_url.scheme_type != GLOBUS_URL_SCHEME_HTTPS  ) {
  printf ("can not copy from %s - wrong prot\n", source_URL);
  return (-1);
}
globus_gass_copy_handleattr_init  (&gass_copy_handleattr);
globus_gass_copy_attr_init        (&source_gass_copy_attr);

globus_ftp_client_handleattr_init (&ftp_handleattr);
globus_io_fileattr_init           (&io_attr);

globus_gass_copy_attr_set_io      (&source_gass_copy_attr, &io_attr);
                                   &io_attr);
globus_gass_copy_handleattr_set_ftp_attr
                                  (&gass_copy_handleattr,
                                   &ftp_handleattr);
globus_gass_copy_handle_init      (&gass_copy_handle,
                                   &gass_copy_handleattr);

if (source_url.scheme_type == GLOBUS_URL_SCHEME_GSIFTP ||
    source_url.scheme_type == GLOBUS_URL_SCHEME_FTP    ) {
  globus_ftp_client_operationattr_init (&source_ftp_attr);
  globus_gass_copy_attr_set_ftp (&source_gass_copy_attr,
                                 &source_ftp_attr);
}
else {
  globus_gass_transfer_requestattr_init (&source_gass_attr,
                                         source_url.scheme);
  globus_gass_copy_attr_set_gass(&source_gass_copy_attr,
           &source_gass_attr);
}

output_file = globus_libc_open ((char*) target,
           O_WRONLY | O_TRUNC | O_CREAT,
           S_IRUSR  | S_IWUSR | S_IRGRP |
           S_IWGRP);
if ( output_file == -1 ) {
  printf ("Could not open the file \"%s\"\n", target);
  return (-1);
}
/* convert stdout to be a globus_io_handle */
if ( globus_io_file_posix_convert (output_file, 0,
                                   &dest_io_handle)
     != GLOBUS_SUCCESS) {
  printf ("Error converting the file handle\n");
  return (-1);
}

result = globus_gass_copy_register_url_to_handle (
         &gass_copy_handle, (char*)source_URL,
         &source_gass_copy_attr, &dest_io_handle,
         my_callback, NULL);
if ( result != GLOBUS_SUCCESS ) {
  printf ("error: %s\n", globus_object_printable_to_string
          (globus_error_get (result)));
  return (-1);
}
globus_url_destroy  (&source_url);
return (0);
}
```

Copy a File: CoG/RFT

```
package org.globus.ogsa.gui;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.net.URL;
import java.util.Date;
import java.util.Vector;
import javax.xml.rpc.Stub;
import org.apache.axis.message.MessageElement;
import org.apache.axis.utils.XMLUtils;
import org.globus.*
import org.gridforum.ogsi.*
import org.gridforum.ogsi.holders.TerminationTimeTypeHolder;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

public class RFTClient {
 public static void copy (String source_url, String target_url) {
  try {
   File requestFile      = new File (source_url);
   BufferedReader reader = null;
   try {
    reader = new BufferedReader (new FileReader (requestFile));
   } catch (java.io.FileNotFoundException fnfe) { }
   Vector requestData = new Vector ();
   requestData.add (target_url);
   TransferType[] transfers1      = new TransferType[transferCount];
   RFTOptionsType multirftOptions = new RFTOptionsType ();

   multirftOptions.setBinary          (Boolean.valueOf  (
         (String)requestData.elementAt (0)).booleanValue ());
   multirftOptions.setBlockSize       (Integer.valueOf  (
         (String)requestData.elementAt (1)).intValue     ());
   multirftOptions.setTcpBufferSize   (Integer.valueOf  (
         (String)requestData.elementAt (2)).intValue     ());
   multirftOptions.setNotpt           (Boolean.valueOf  (
         (String)requestData.elementAt (3)).booleanValue ());
   multirftOptions.setParallelStreams (Integer.valueOf  (
         (String)requestData.elementAt (4)).intValue     ());
   multirftOptions.setDcau(Boolean.valueOf(
          (String)requestData.elementAt (5)).booleanValue ());

  int i = 7;
  for (int j = 0; j < transfers1.length; j++)
  {
   transfers1[j] = new TransferType ();

   transfers1[j].setTransferId     (j);
   transfers1[j].setSourceUrl      ((String)requestData.elementAt (i++));
   transfers1[j].setDestinationUrl ((String)requestData.elementAt (i++));
   transfers1[j].setRftOptions     (multirftOptions);
  }

TransferRequestType transferRequest = new TransferRequestType ();
transferRequest.setTransferArray (transfers1);

int concurrency = Integer.valueOf
                  ((String)requestData.elementAt(6)).intValue();

if (concurrency > transfers1.length)
{
  System.out.println ("Concurrency should be less than the number"
                      "of transfers in the request");
  System.exit (0);
}
transferRequest.setConcurrency (concurrency);

TransferRequestElement requestElement = new TransferRequestElement ();
requestElement.setTransferRequest (transferRequest);

ExtensibilityType extension = new ExtensibilityType ();
extension = AnyHelper.getExtensibility (requestElement);

OGSIServiceGridLocator factoryService = new OGSIServiceGridLocator ();
Factory factory = factoryService.getFactoryPort (new URL (source_url));
GridServiceFactory gridFactory = new GridServiceFactory (factory);

LocatorType locator = gridFactory.createService (extension);
System.out.println ("Created an instance of Multi-RFT");

MultiFileRFTDefinitionServiceGridLocator loc
                        = new MultiFileRFTDefinitionServiceGridLocator();
RFTPortType rftPort = loc.getMultiFileRFTDefinitionPort (locator);
  ((Stub)rftPort)._setProperty (Constants.AUTHORIZATION,
                                NoAuthorization.getInstance());
  ((Stub)rftPort)._setProperty (GSIConstants.GSI_MODE,
                                GSIConstants.GSI_MODE_FULL_DELEG);
  ((Stub)rftPort)._setProperty (Constants.GSI_SEC_CONV,
                                Constants.SIGNATURE);
  ((Stub)rftPort)._setProperty (Constants.GRIM_POLICY_HANDLER,
                                new IgnoreProxyPolicyHandler ());

  int requestid = rftPort.start ();
  System.out.println ("Request id: " + requestid);

 }
 catch (Exception e)
 {
  System.err.println (MessageUtils.toString (e));
 }
}
}
```


Copy a File: GAT/C

```
#include <GAT.h>

GATResult RemoteFile_GetFile (GATContext context,
        char const* source_url, char const* target_url)
{
  GATStatus   status  = 0;
  GATLocation source  = GATLocation_Create (source_url);
  GATLocation target  = GATLocation_Create (target_url);
  GATFile     file    = GATFile_Create (context, source, 0);
  if (source == 0 || target == 0 || file == 0) {
      return GAT_MEMORYFAILURE; }
  if ( GATFile_Copy (file, target, GATFileMode_Overwrite) != GAT_SUCCESS )
  {
    GATContext_GetCurrentStatus    (context, &status);
    return GATStatus_GetStatusCode (status);
  }
  GATFile_Destroy     (&file);
  GATLocation_Destroy (&target);
  GATLocation_Destroy (&source);

  return GATStatus_GetStatusCode (status);
}
```

Copy a File: GAT/C++

```
#include <GAT++.hpp>

GAT::Result RemoteFile::GetFile (GAT::Context context,
                                 std::string source_url,
                                 std::string target_url)
{
  try
  {
    GAT::File file (context, source_url);
    file.Copy      (target_url);
  }
  catch (GAT::Exception const &e)
  {
    std::cerr << "Some error: " << e.what() << std::endl;
    return e.Result();
  }
  return GAT_SUCCESS;
}
```


Copy a File: GAT/C++

```
#include <GAT++.hpp>

GAT::Result RemoteFile::GetFile (GAT::Context context,
                                 std::string source_url,
                                 std::string target_url)
{
  try
  {
    GAT::File file (context, source_url);
    file.Copy      (target_url);
  }
  catch (GAT::Exception const &e)
  {
    std::cerr << "Some error: " << e.what() << std::endl;
    return e.Result();
  }
  return GAT_SUCCESS;
}
```

Copy a File: GAT/C

```
#include <GAT.h>

GATResult RemoteFile_GetFile (GATContext context,
        char const* source_url, char const* target_url)
{
  GATStatus   status  = 0;
  GATLocation source  = GATLocation_Create (source_url);
  GATLocation target  = GATLocation_Create (target_url);
  GATFile     file    = GATFile_Create (context, source, 0);
  if (source == 0 || target == 0 || file == 0) {
      return GAT_MEMORYFAILURE; }
  if ( GATFile_Copy (file, target, GATFileMode_Overwrite) != GAT_SUCCESS )
  {
    GATContext_GetCurrentStatus    (context, &status);
    return GATStatus_GetStatusCode (status);
  }
  GATFile_Destroy     (&file);
  GATLocation_Destroy (&target);
  GATLocation_Destroy (&source);

  return GATStatus_GetStatusCode (status);
}
```


GAT Solution

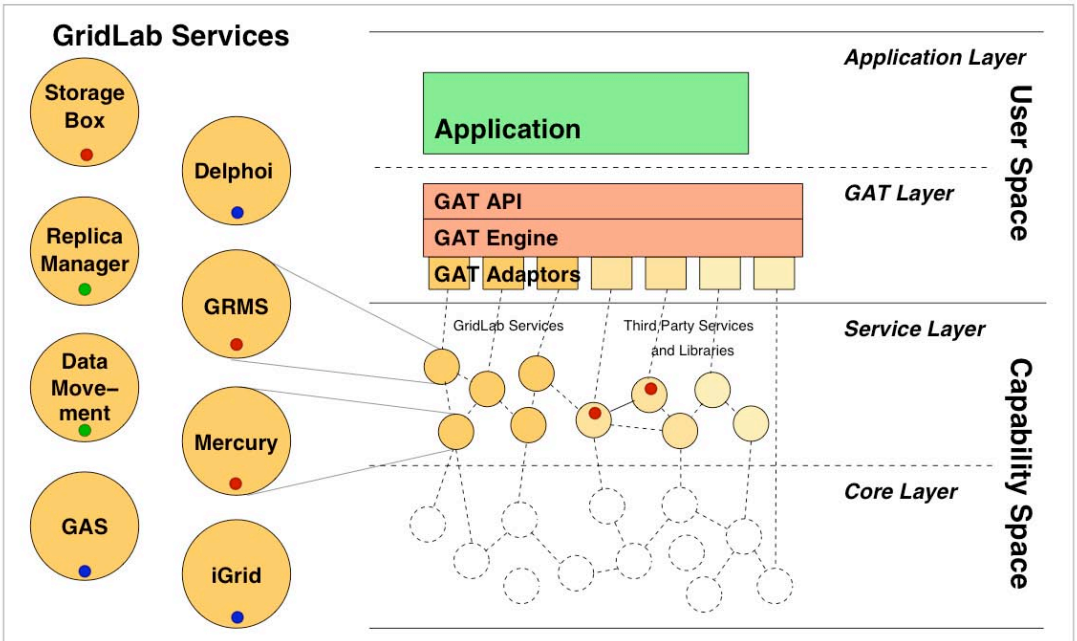
GAT API layer between applications and the grid infrastructure:

- Higher level than existing grid APIs, hide complexity, abstract grid functionality through application oriented APIs
- Insulate against
  - Rapid evolution of grid infrastructure
  - State of Grid deployment
- Choose between different grid infrastructures

Grid Application Toolkit

- Standard API and Toolkit for developing portable Grid applications independently of the underlying Grid infrastructure and available services
- Implements the GAT-API
  - Used by applications (different languages)
- GAT Adaptors
  - Connect to capabilities/services
  - Implement well defined CPI (mirrors GAT-API)
  - Interchangeable adaptors can be loaded/switched at runtime

GridLab Architecture



GAT API Subsystems

File Subsystem		
GATFile	GATLogicalFile	
GATEndpoint	GATPipeListener	GATPipe
Monitoring and Event Subsystem		
GATRequestListener	GATRequestNotifier	GATAction
GATMetricListener	GATMetric	GATMetricEvent
Information Exchange Subsystem		
GATAdvertisable	GATAdvertService	
Resource Management Subsystem		
GATSoftwareDescription	GATResourceDescription	GATResource
GATJobDescription	GATResourceBroker	GATReservation
GATJob		
Utility Subsystem		
GATSelf	GATContext	GATSecurityContext
GATStatus	GATPreferences	URL, Time, ...

Examples: Read Remote File

```
try
{
  char data[25];
  GAT::File file (context, source_url);
  file.open  (RD_ONLY);
  file.seek  (100, SEEK_SET);
  file.read  (data, sizeof(data));
  file.close ();
}
catch (GAT::Exception const &e)
{
  std::cerr << "Some error: " << e.what() << std::
    endl;
  return e.Result();
}
```

Examples: Read Logical File

```
try
{
  char data[25];
  GAT::LogicalFile       logical_file (context, name);
  list<GAT::File> files = logical_file.get_files ();

  files[0].open   (RD_ONLY);
  files[0].seek   (100, SEEK_SET);
  files[0].read   (data, sizeof(data));
  files[0].close  ();
}
catch (GAT::Exception const &e)
{
  std::cerr << "Some error: " << e.what() << std::endl;
  return e.Result();
}
```


Example: Spawn a Subtask

```
GAT::Table sdt;   sdt.add ("location",     "/bin/date");
GAT::Table hdt;   hdt.add ("machine.type", "i686");

GAT::SoftwareDescription          sd  (sdt);
GAT::HardwareResourceDescription  hrd (hdt);

GAT::JobDescription jd (context, sd, hrd);
GAT::ResourceBroker rb (context, prefs);

GAT::Job j = rb.submit (jd);
```

Implementation

- C/C++/Java versions fully implemented
- Basic Python and .NET versions
- Perl, Fortran to follow
- Focus: portability, lightness, flexibility, adaptivity

Credits

- A few of these slides were created by Alain Roy for the 2004 Grid School, Brownsville.
- Alain created some from slides from the Globus project.

Links

Infrastructure:

- Cactus Framework: http://www.cactuscode.org
- Grid Application Toolkit: http://www.gridlab.org
- GriKSL: http://www.griksl.org

Applications:

- GridChem: http://www.gridchem.org
- UCoMS: http://www.ucoms.org