Introduction to Grid Computing

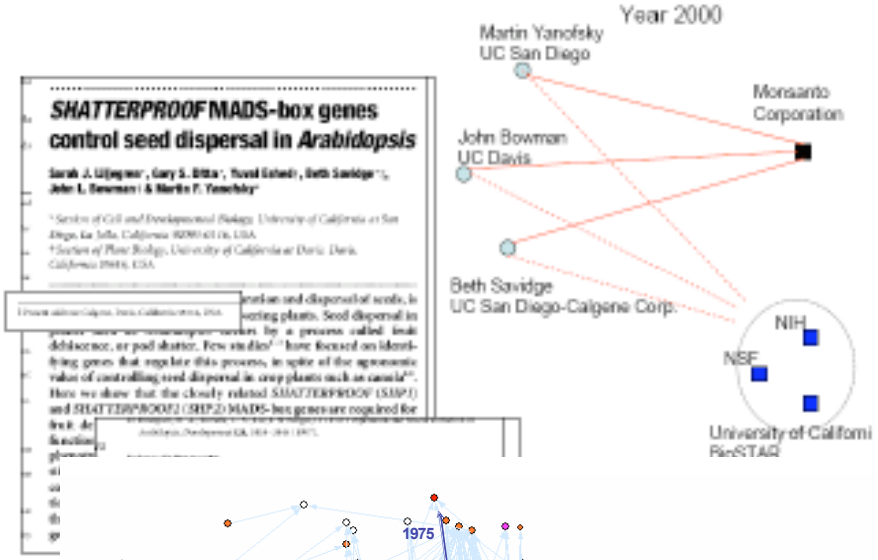
Health Grid 2008
University of Chicago Gleacher Center
Chicago, IL USA
June 2, 2008

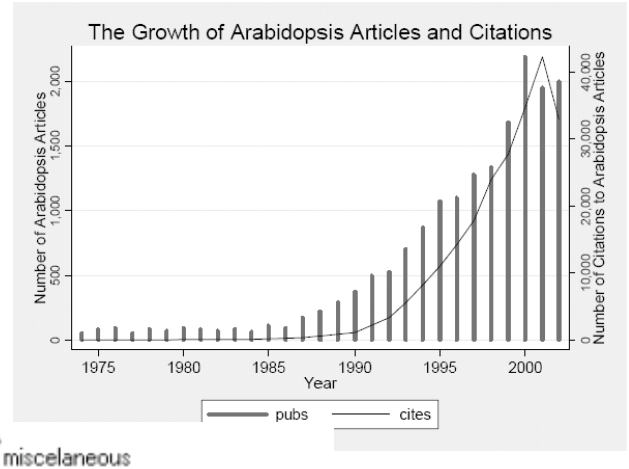
Open Science Grid

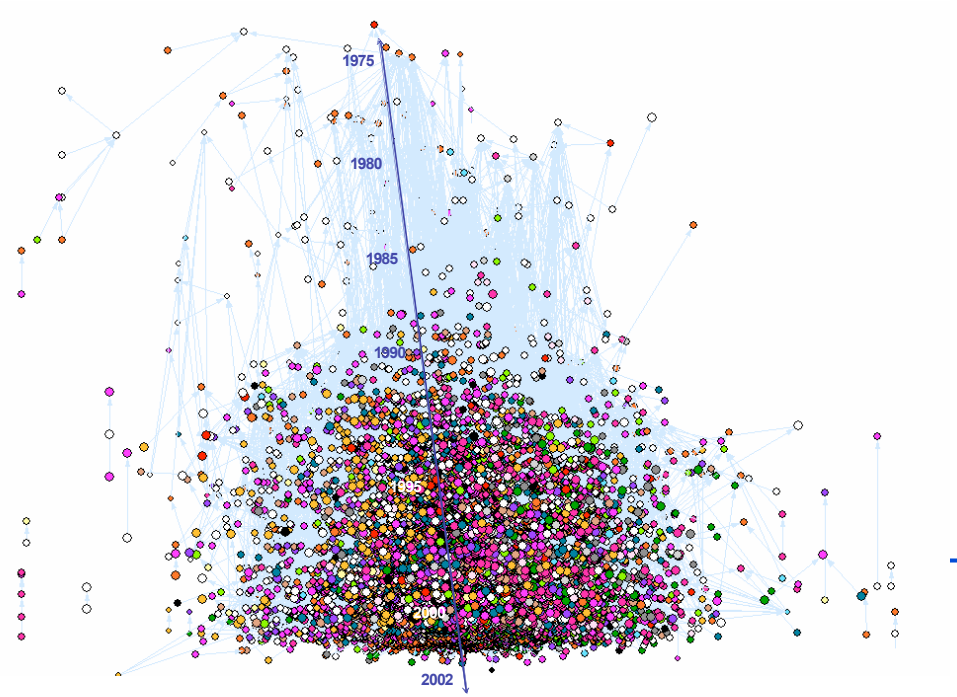
Introduction to Grid Computing Tutorial Outline

I. Motivation and Grid Architecture
II. Grid Examples from Life Sciences
III. Grid Security
IV. Job Management: Running Applications
V. Data Management
VI. Open Science Grid and TeraGrid
VII. Workflow on the Grid
VIII. Next steps: Learning more, getting started

Scaling up Science: Biological Research Citation Network Analysis







Work of James Evans, University of Chicago, Department of Sociology

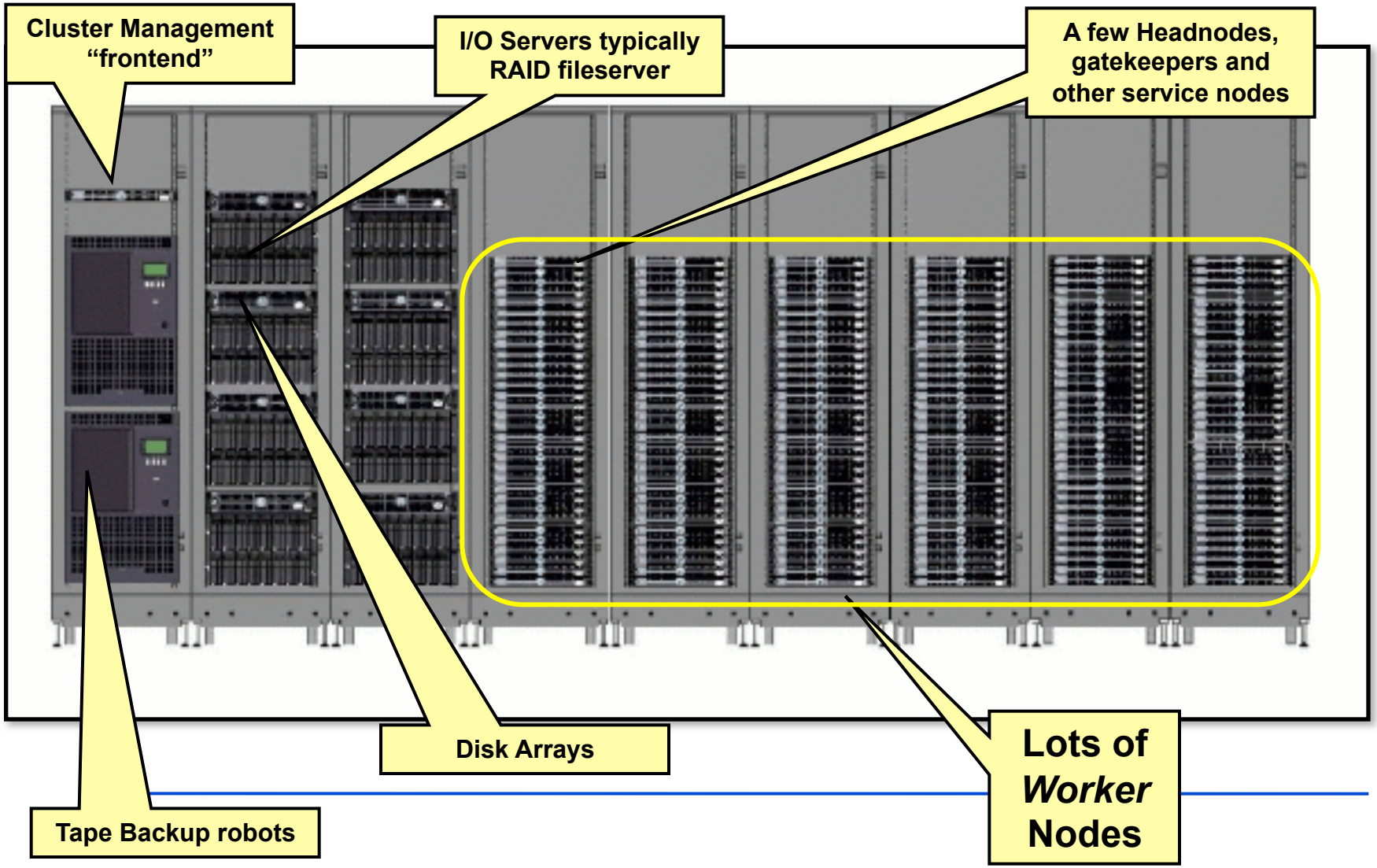
Scaling up the analysis

- Query and analysis of 25+ million citations
- Work started on desktop workstations
- Queries grew to month-long duration
- With data distributed across
  U of Chicago/OSG TeraPort **cluster**:
  - 50 (faster) CPUs gave 100 X speedup
  - **Far more methods and hypotheses can be tested!**
- Higher *throughput* and *capacity* enables *deeper analysis* and *broader community access*.

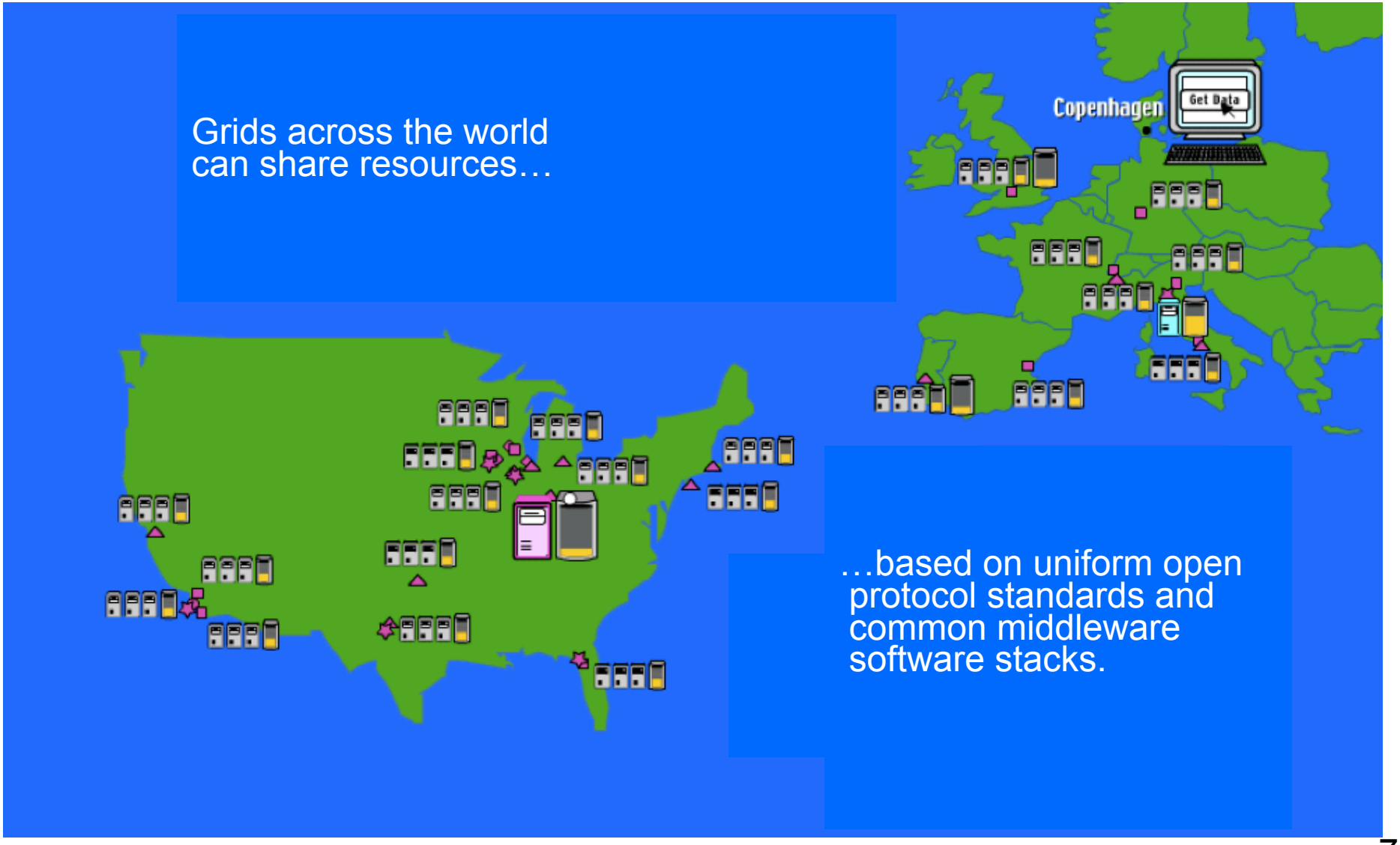
What makes this possible?

- High performance computing
  - Once the domain of costly, specialized supercomputers (many architectures – programming was difficult)
  - Now unified around a smaller number of popular models – notably, message passing
  - "tightly coupled" computing – message passing and multicore: passes data via memory and/or messages
- High throughput computing
  - Running lots of ordinary programs in parallel
  - Loosely-coupled: passes data via file exchange
  - Highly accessible to scientists – needs little programming expertise
  - "Scripting" enables diverse execution models
- **Supercomputing via commodity clusters**

Computing clusters have commoditized supercomputing



Grids Provide Global Resources To Enable e-Science

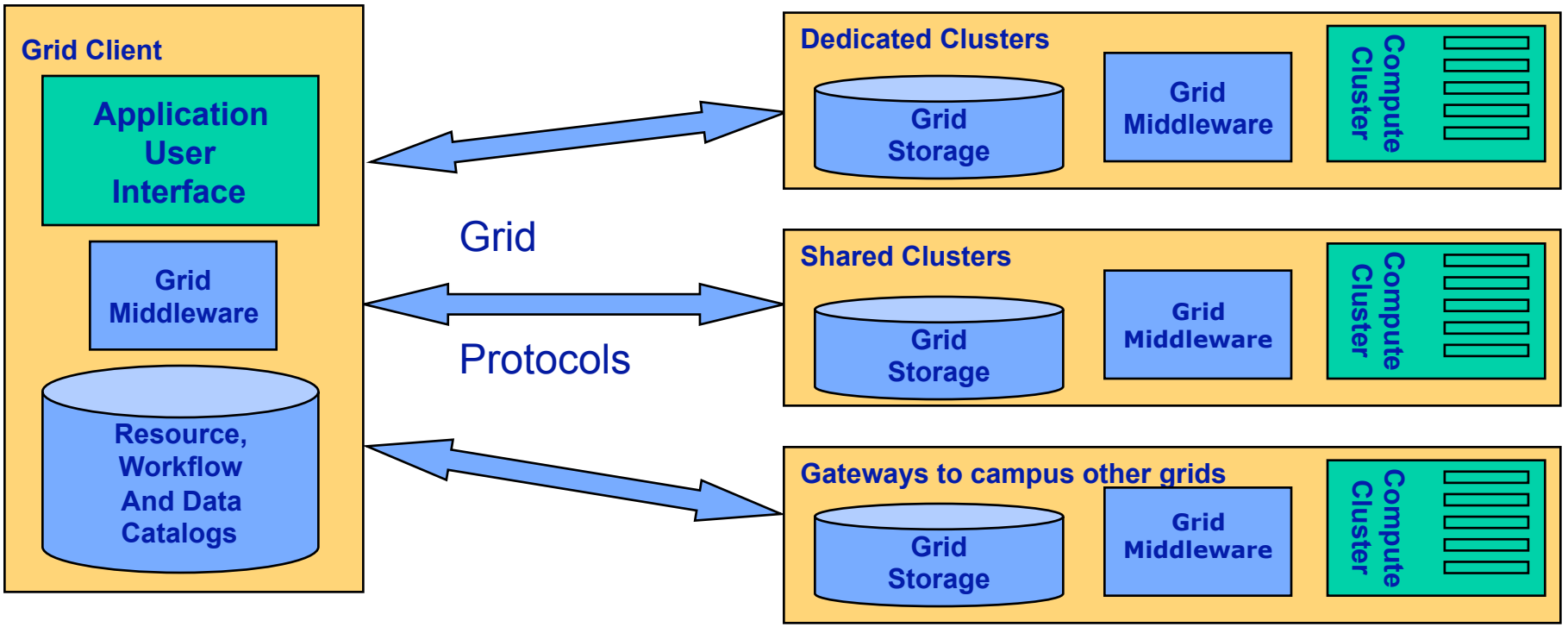


What is a Grid?

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

Grids are composed of distributed **clusters**…



…and a set of services provided by middleware toolkits and protocols

- Security to control access and protect communication
- Directory to locate grid sites and services
- Uniform interface to computing sites
- Facility to maintain and schedule queues of work
- Fast and secure data set movers
- Directories to track where datasets live

Initial Grid driver: High Energy Physics

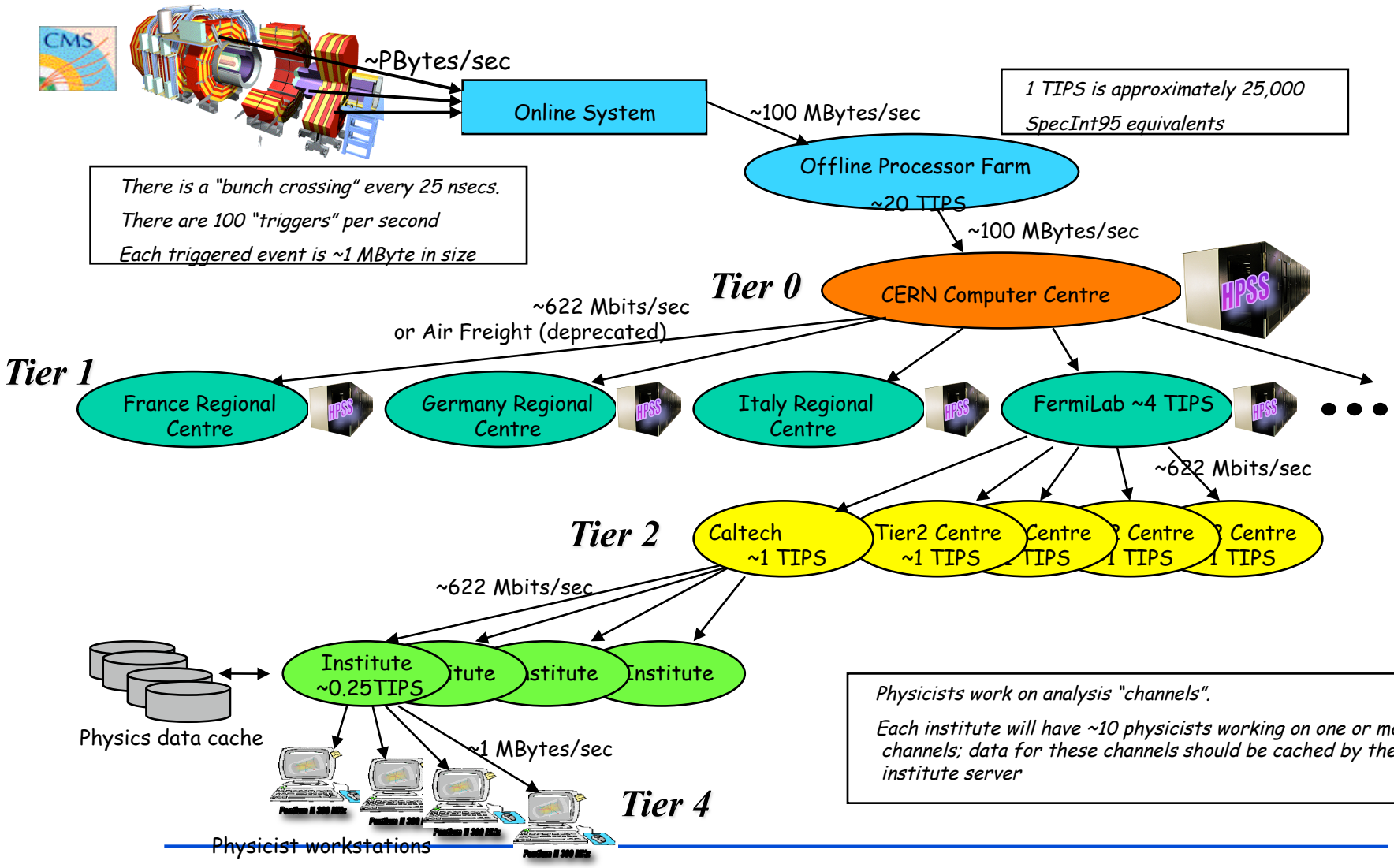
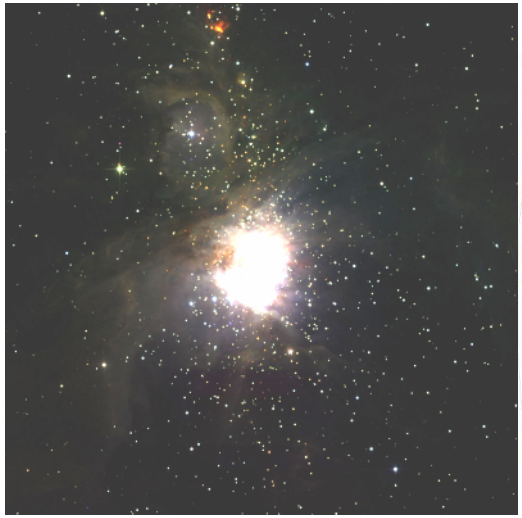


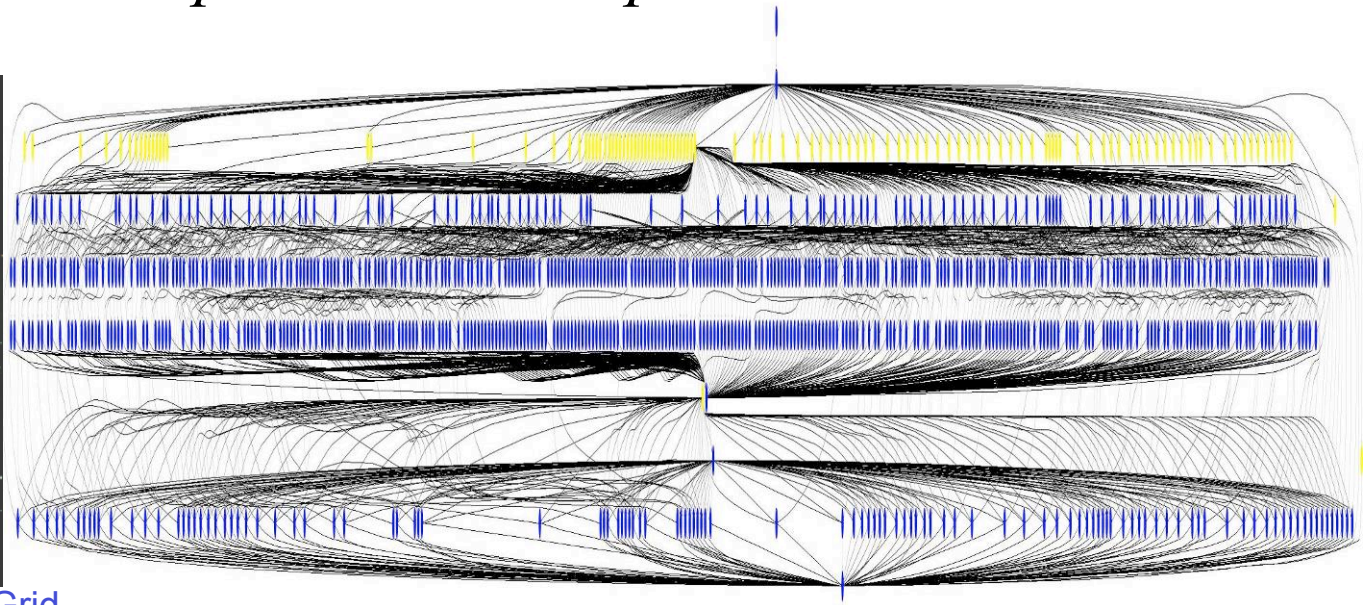
Image courtesy Harvey Newman, Caltech

Grids can process vast datasets.

- Many HEP and Astronomy experiments consist of:
  - Large datasets as inputs (find datasets)
  - "Transformations" which work on the input datasets (process)
  - The output datasets (store and publish)
- The emphasis is on the sharing of the large datasets
- Programs when *independent* can be *parallelized*.



Mosaic of M42 created on TeraGrid

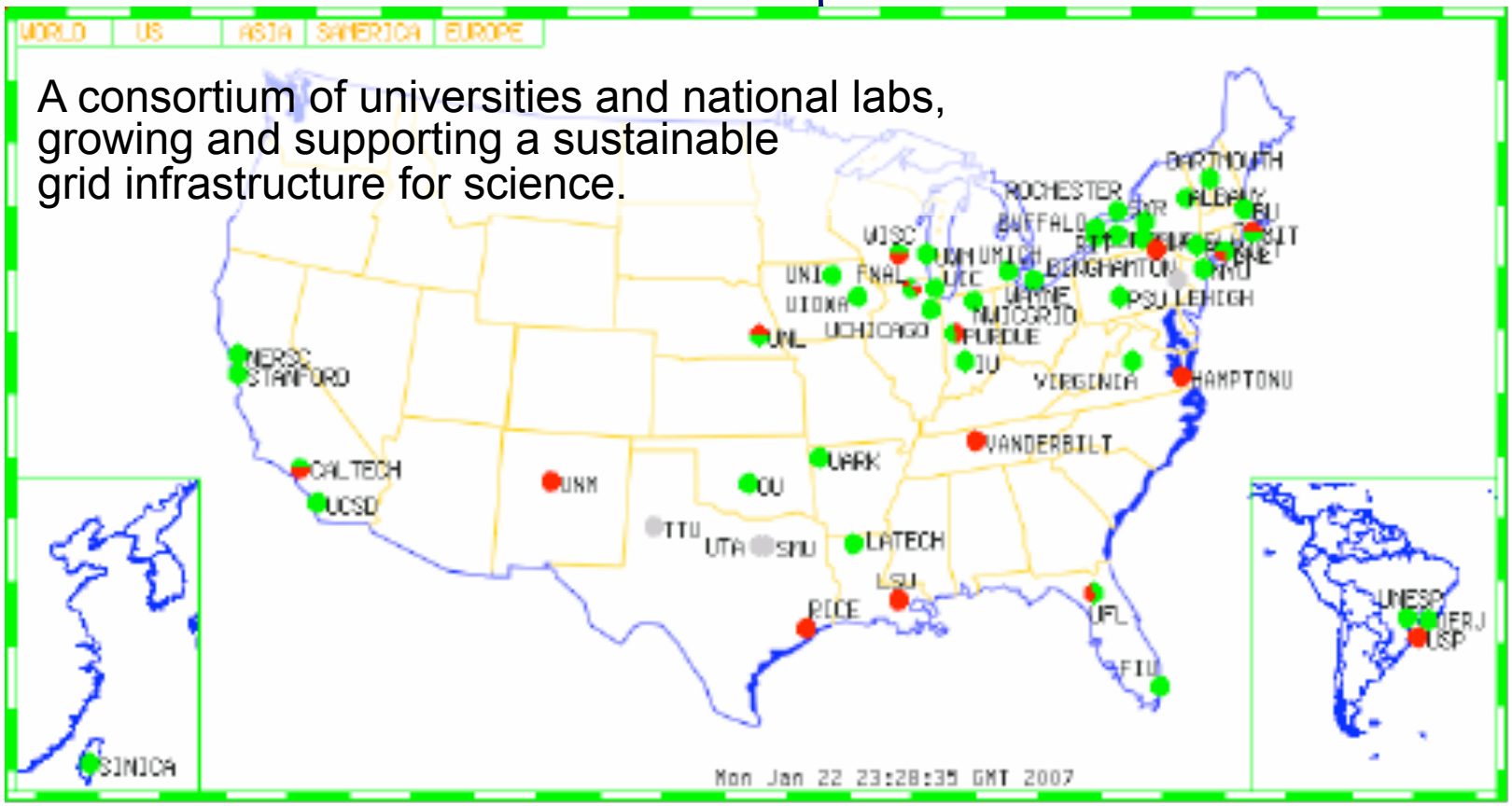


= Data Transfer

= Compute Job

Montage Workflow: ~1200 jobs, 7 levels
NVO, NASA, ISI/Pegasus - Deelman et al.

Open Science Grid (OSG) provides shared computing resources for a broad set of disciplines



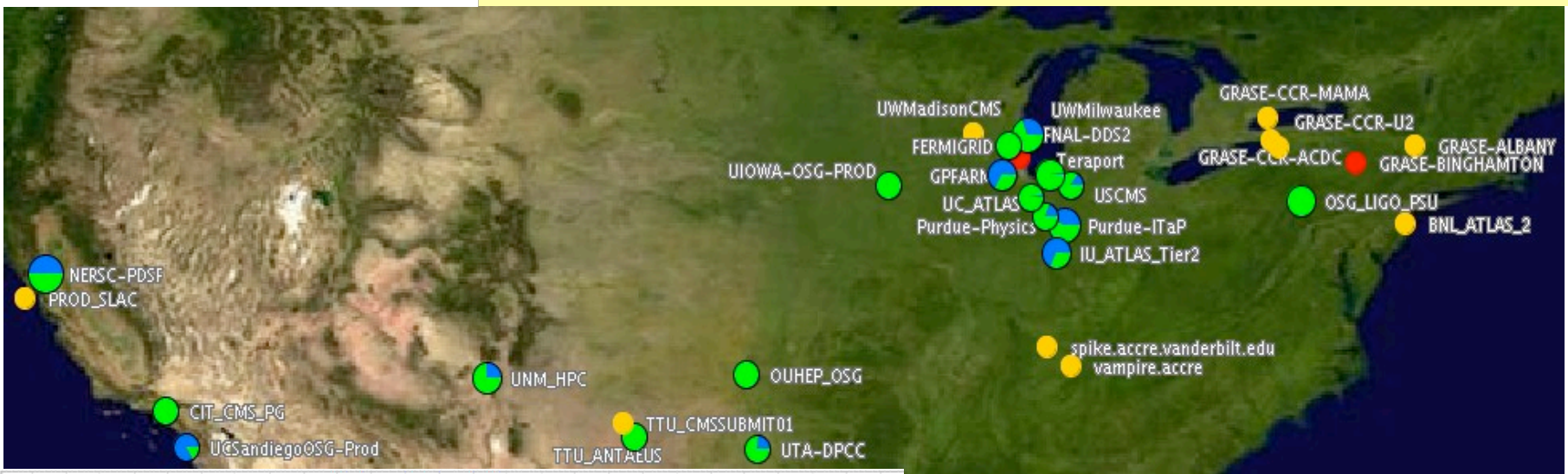
- OSG focuses on general services, operations, and end-to-end performance
- Composed of a large number (>75 and growing) of shared computing facilities, or "sites"
- Uses advanced networking from all available high-performance research networks

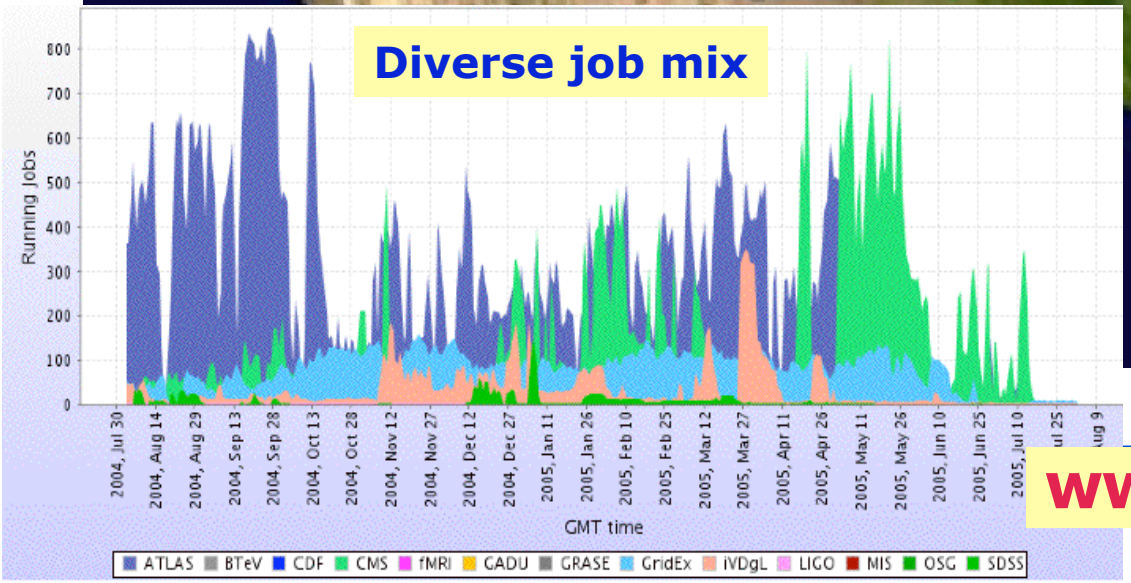
Open Science Grid

http://www.opensciencegrid.org

Open Science Grid

- 70 sites (20,000 CPUs) & growing
- 400 to >1000 concurrent jobs
- Many applications + CS experiments; includes long-running production operations





www.opensciencegrid.org

TeraGrid provides vast resources via a number of huge computing facilities.

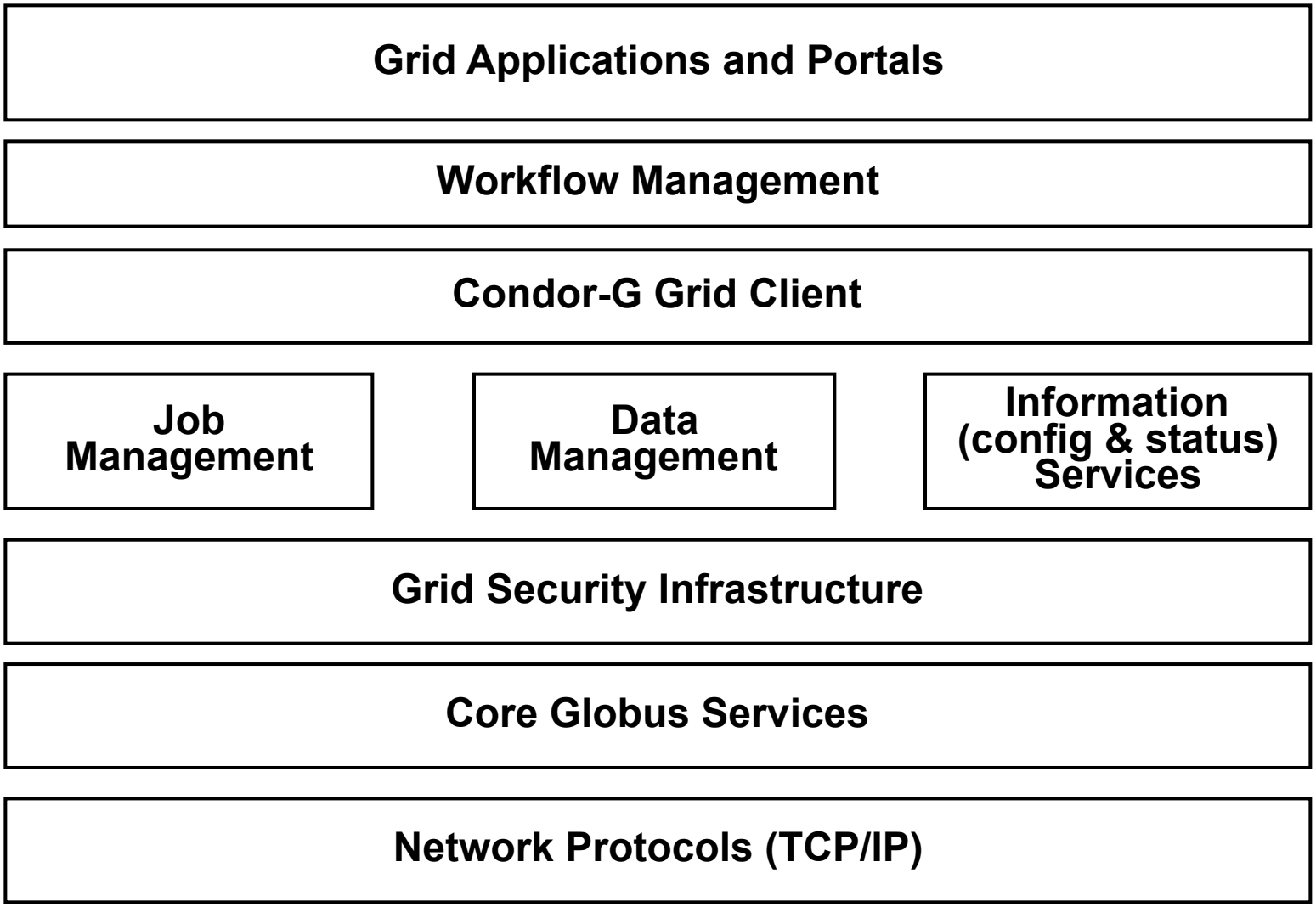
Virtual Organizations (VOs)

- Groups of organizations that use the Grid to share resources for specific purposes
- Support a single community
- Deploy compatible technology and agree on working policies
  - Security policies - difficult
- Deploy different network accessible services:
  - Grid Information
  - Grid Resource Brokering
  - Grid Monitoring
  - Grid Accounting

Grid Software: Globus and Condor

- Condor provides both *client* & *server* scheduling
  - Condor-G: an agent to queue, schedule and manage work submission
- Globus Toolkit (GT) provides core middleware
  - Client tools which you can use from a command line
  - APIs (scripting languages, C, C++, Java, …) to build your own tools, or use direct from applications
  - Web service interfaces
  - Higher level tools built from these basic components, e.g. Reliable File Transfer (RFT)

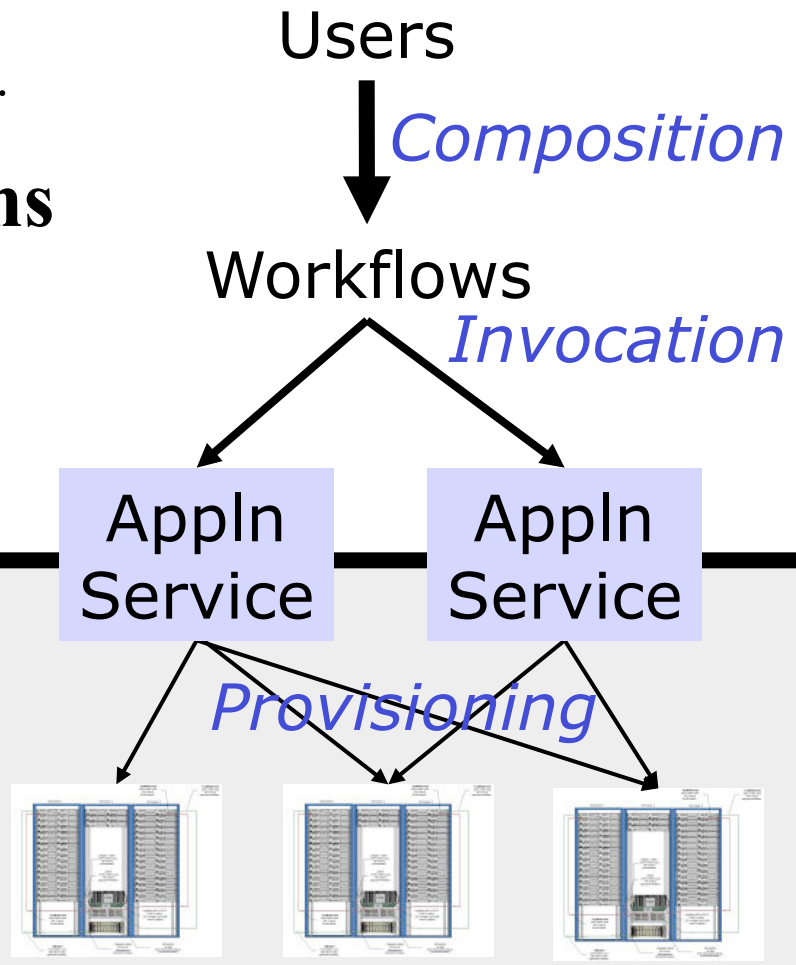
Grid software "stack"



Grid architecture is evolving to a Service-Oriented approach.

...but this is beyond our workshop's scope.
See "Service-Oriented Science" by Ian Foster.

- Service-oriented **applications**
  - Wrap applications as services
  - Compose applications into workflows
- Service-oriented **Grid infrastructure**
  - Provision physical resources to support application workloads



"The Many Faces of IT as Service", Foster, Tuecke, 2005

Introduction to Grid Computing Tutorial Outline

I. Motivation and Grid Architecture
II. Grid Examples from Life Sciences
III. Grid Security
IV. Job Management: Running Applications
V. Data Management
VI. Open Science Grid and TeraGrid
VII. Workflow on the Grid
VIII. Next steps: Learning more, getting started

How to leverage grids in biology and health?

Many biological processes are statistical in nature.

- Simulations of such processes parallelize naturally.

Performance of multiple simulations on OSG – the **Open Science Grid** –
will help to:

- Enhance Accuracy
  - Sampling necessary for calculation of experimental observables
  - Simulate mutants, compare their behavior and compare with experiments
  - Test several force-fields - different force-fields may produce different results
- Describe long timescale processes (μs and longer)
  - Run large number of simulations (increase probability of observation of important events)
  - Test different “alternative” methods

Example: OSG CHARMM Application

Project and Slides are the work of:

Ana Damjanovic (JHU, NIH)

JHU:

Petar Maksimovic

Bertrand Garcia-Moreno

NIH:

Tim Miller

Bernard Brooks

OSG:

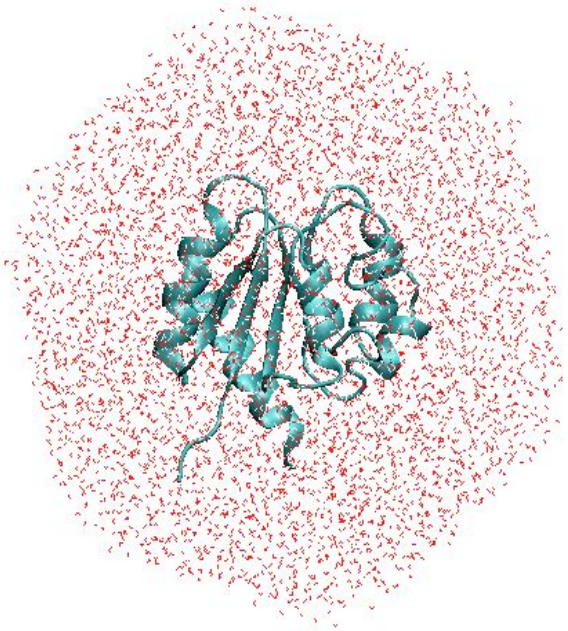
Torre Wenaus and team

CHARMM and MD simulations

CHARMM is one of the most widely used programs for computational modeling, simulations and analysis of biological (macro)molecules

The widest use of CHARMM is for **molecular dynamics** (MD) simulations:

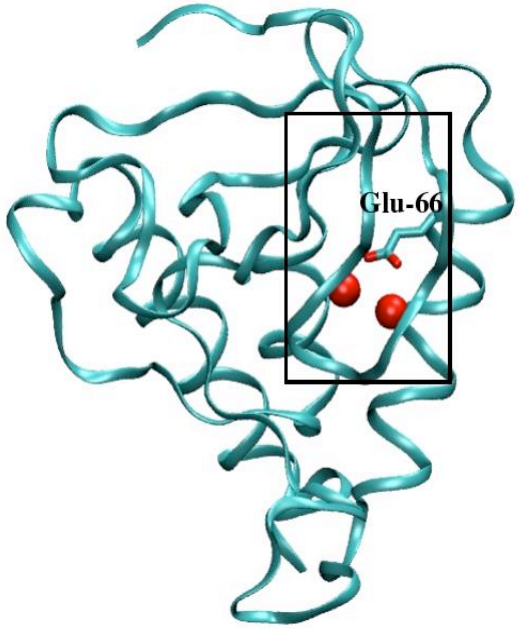
- Atoms are described explicitly
- interactions between atoms are described with an empirical force-field
  * electrostatic, van der Waals
  * bond vibrations, angle bending, dihedrals …
- Newton's equations are solved to describe
  time evolution of the system: timestep 1-2 fs
  typical simulation times: 10-100 ns
- CHARMM has a variety of tools for analysis of MD trajectories



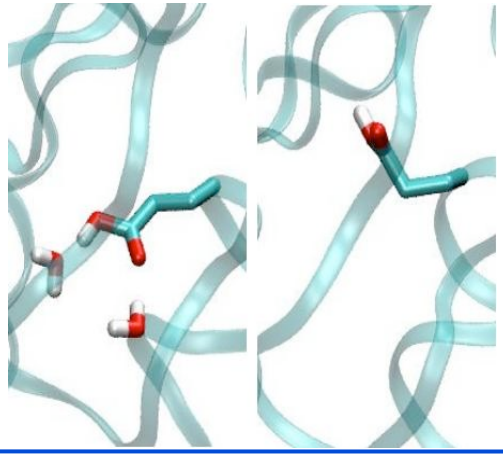
Hydration of the protein interior

- Interior water molecules can play key roles in many biochemical processes such as proton transfer, or catalysis.
- Precise location and numbers of water molecules in protein interiors is not always known from experiments.
- Knowing their location is important for understanding how proteins work, but also

staphylococcal nuclease



Crystallographic structures obtained at different temperatures disagree in the number of observed water molecules.
How many water molecules are in the protein interior?



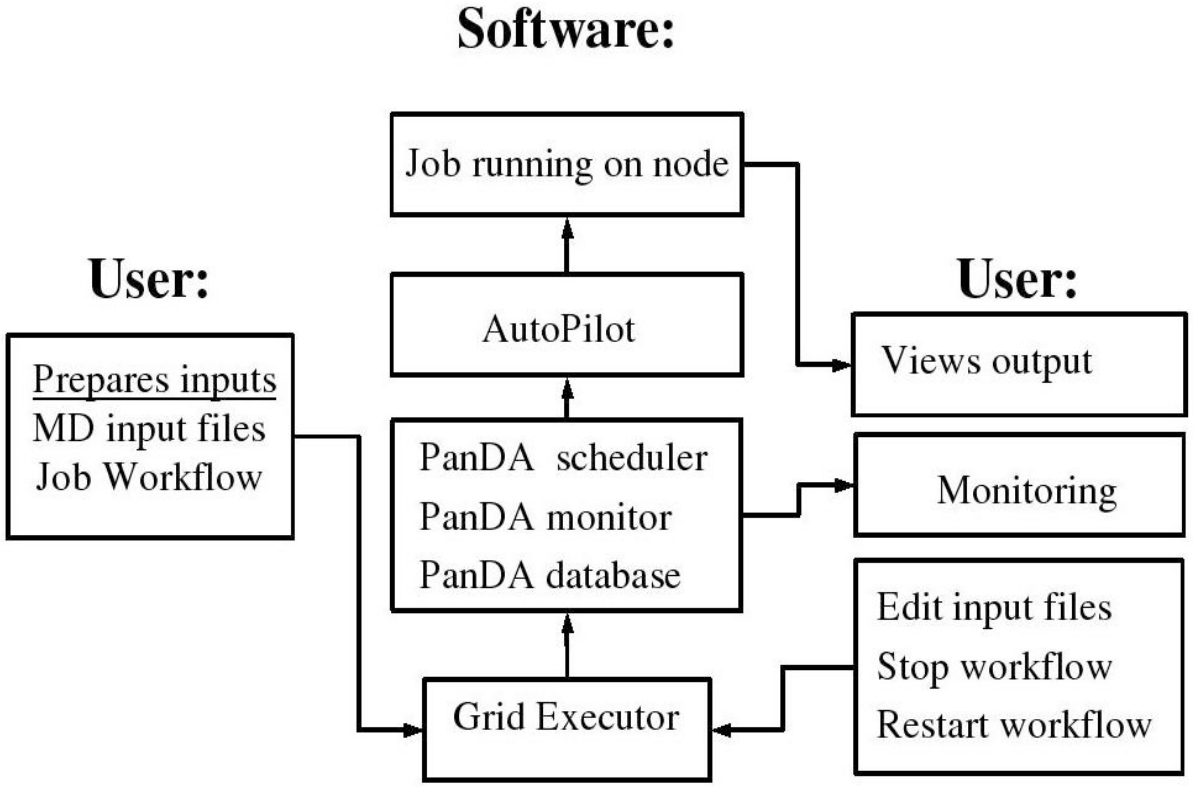
Using traditional HPC resources we performed 10 X 10 ns long MD simulations.

Two conformations, each different hydration pattern

Not enough statistics!

Use OSG to run lots of simulations with different initial velocities.

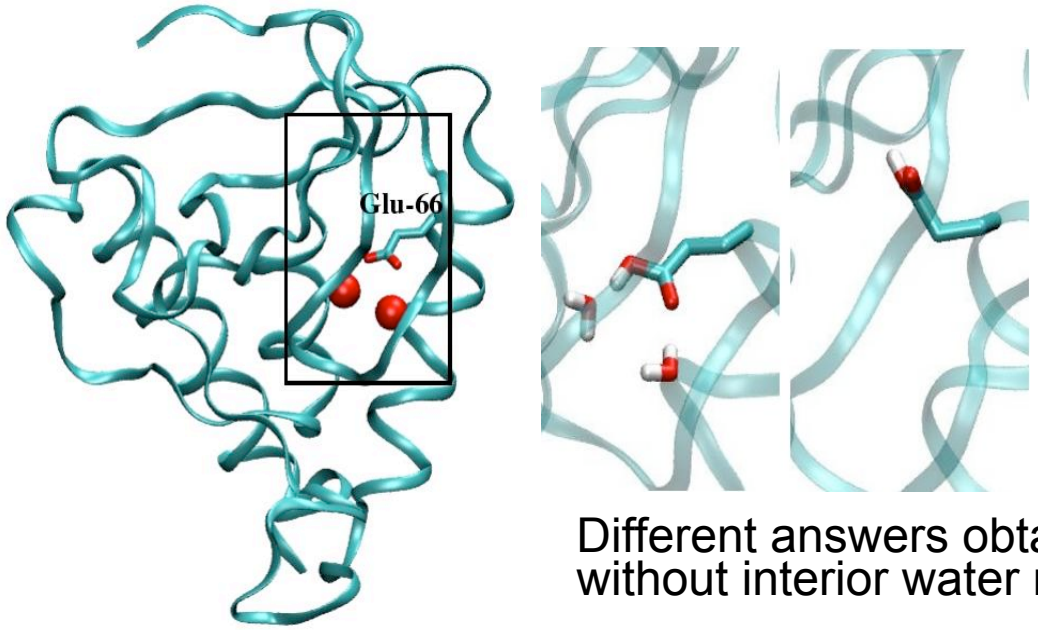
Running jobs on the OSG



- Manual submission and running of a large number of jobs can be time-consuming and lead to errors
- OSG personnel worked with scientists to get this scheme running
- PanDA was developed for ATLAS, and is being evolved into OSG-WMS

Accomplishments with use of OSG

staphylococcal nuclease



2 initial structures X 2 methods
each 40 X 2 ns long simulations
Total: 160 X 2ns.
Total usage: 120K cpu hours

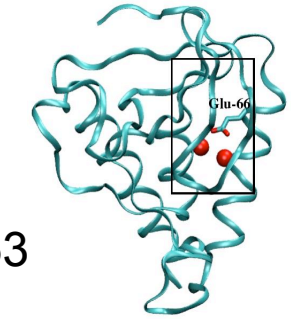
Interior water molecules can influence protein conformations

Different answers obtained if simulations started with and without interior water molecules

Longer runs are needed to answer “how many water molecule are in the protein?”

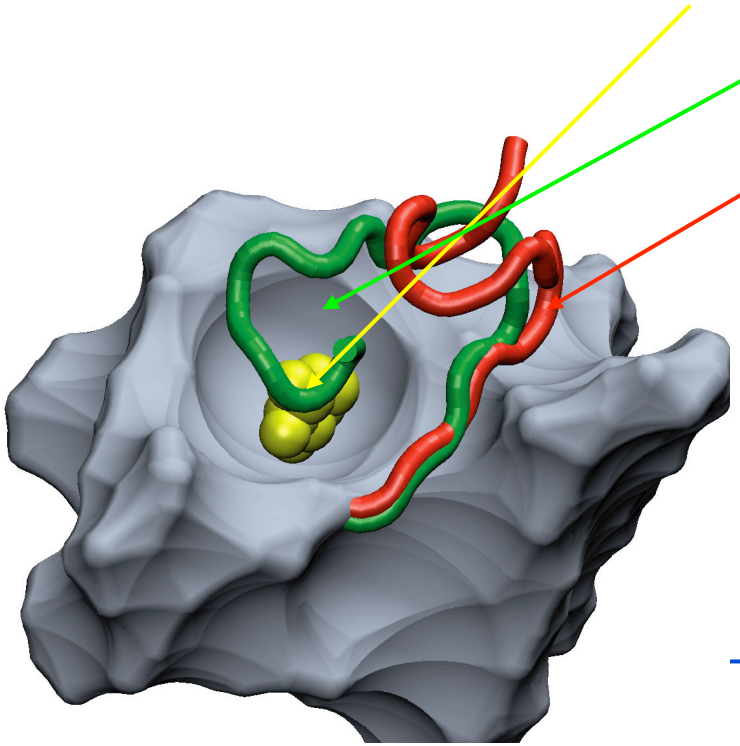
Parallel CHARMM is key to running longer simulations (as simulation time is reduced). OSG is exploring and testing ways to execute **parallel MPI applications**.

Plans for near term future



- **Hydration of interior of staphylococcal nuclease:**
  - answer previous question: (80 X 8 ns = 153K cpu hours)
  - test new method for conformational search, SGLD (80 X 8 ns = 153)

- **Conformational rearrangements and effects of mutations in AraC protein.**
  - when sugar **arabinose** is present, "**arm**", gene expression on
  - when sugar **arabinose** is absent, "**arm**", gene expression off



Computational requirements:

- Wild type with and without arabinose (50 X 10 ns)
- F15V with and without arabinose (50 X 10 ns)

Total: 100 X 10 ns = 240K cpu hours

Additional test simulations (100 X 25 X 1 ns)

Total: 2500 X 1ns = 600K cpu hours

Provide feedback to experiments

Long term needs

Study conformational rearrangements in other proteins

- Conformational rearrangements are at the core of key biochemical processes such as regulation of enzymatic and genetic activity.
- Understanding of such processes is pharmaceutically relevant.
- Such processes usually occur on timescales of μs and longer, are not readily sampled in MD simulations.
- Experimentally little is know about the mechanisms of these processes.
- Poorly explored territory, will be “hot” for the next 10 years
    - Use brute force approach
    - Test performance of different methods
    - Test different force fields
    - Study effects of mutations -> provide feedback to experiments

Impact on scientific community

• CHARMM on OSG is still in testing and development stage so the user pool is small

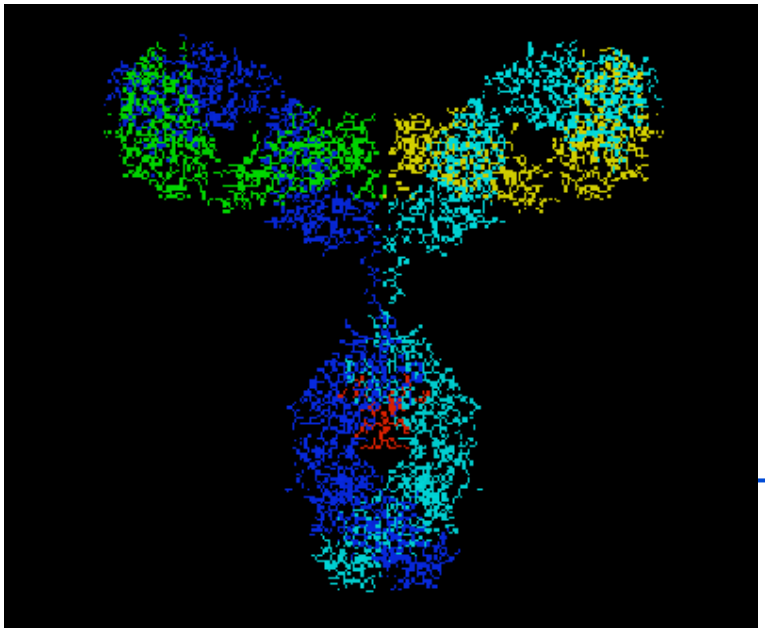
• Potential is great - there are 1,800 registered CHARMM users

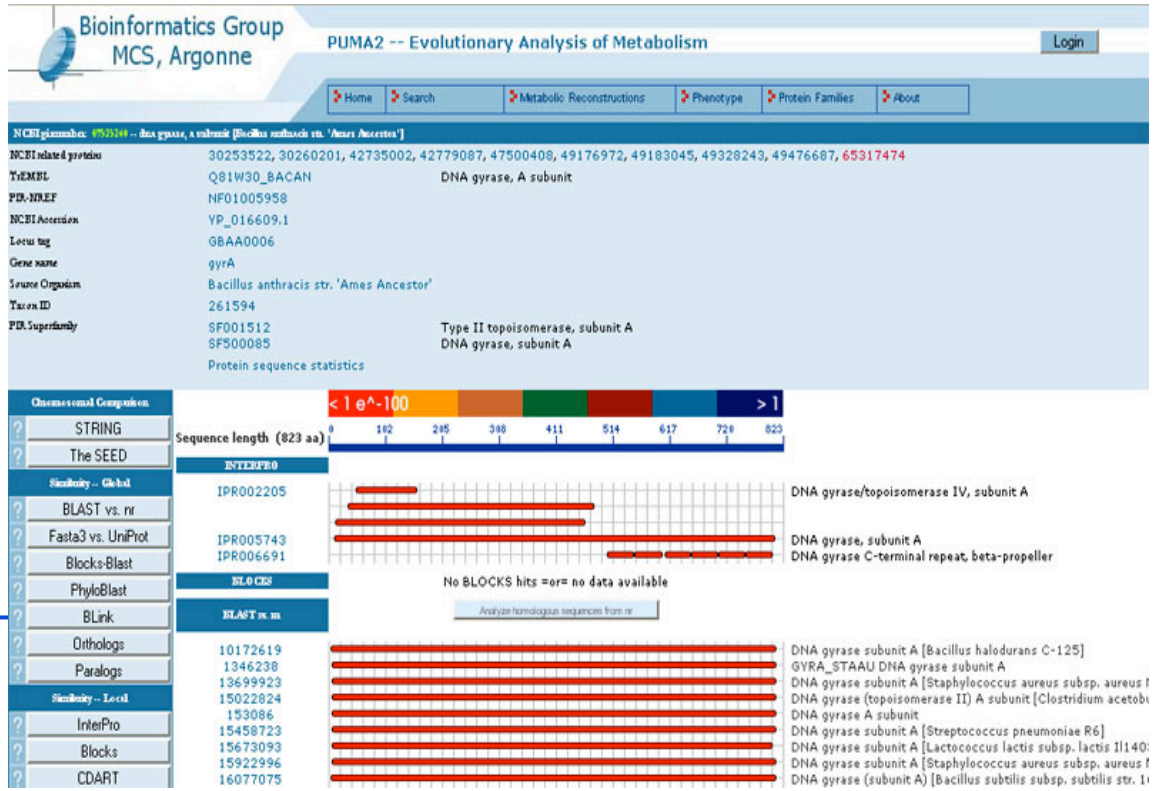
• bring OSG computing to **hundreds** of other CHARMM users
  – give resources to small groups
  – do more science by harvesting unused OSG cycles
  – provide job management software

• "Recipe" used here developed for CHARMM, but can be easily extended to other MD programs (AMBER, NAMD, GROMACS...)

Genome Analysis and Database

- Runs across TeraGrid and OSG. Used VDL and Pegasus workflow & provenance.
- Scans public DNA and protein databases for new and newly updated genomes of different organisms and runs BLAST, Blocks, Chisel. 1200 users of resulting DB.
- On OSG at the peak used >600CPUs,17,000 jobs a week.

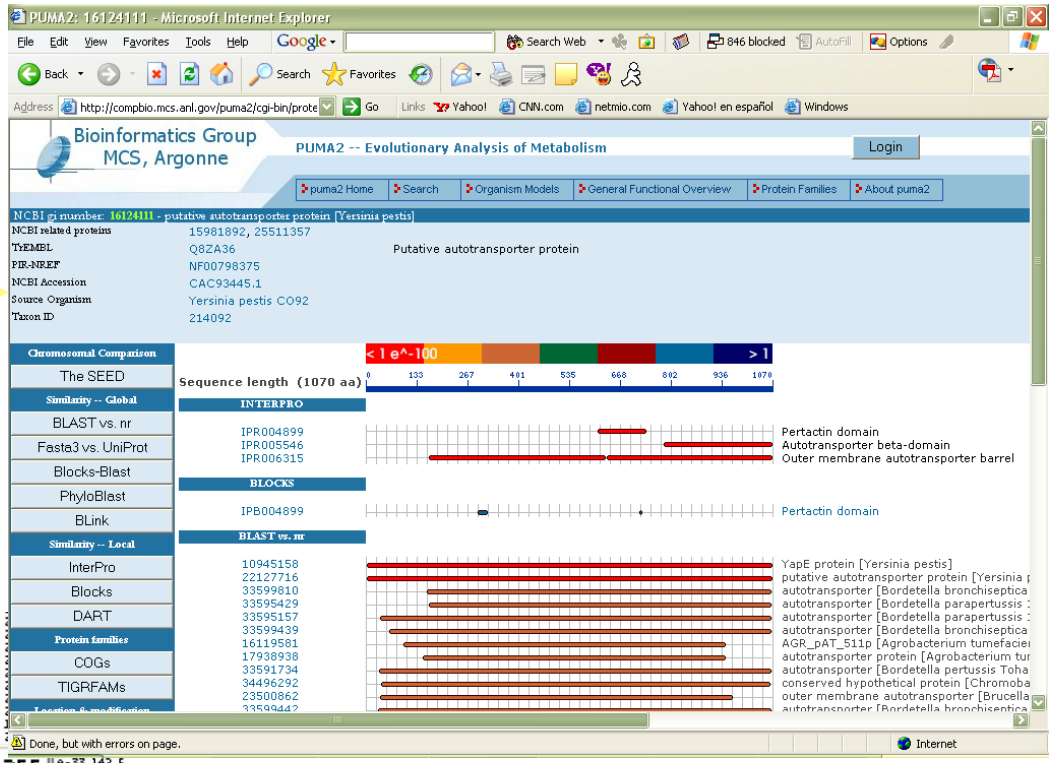


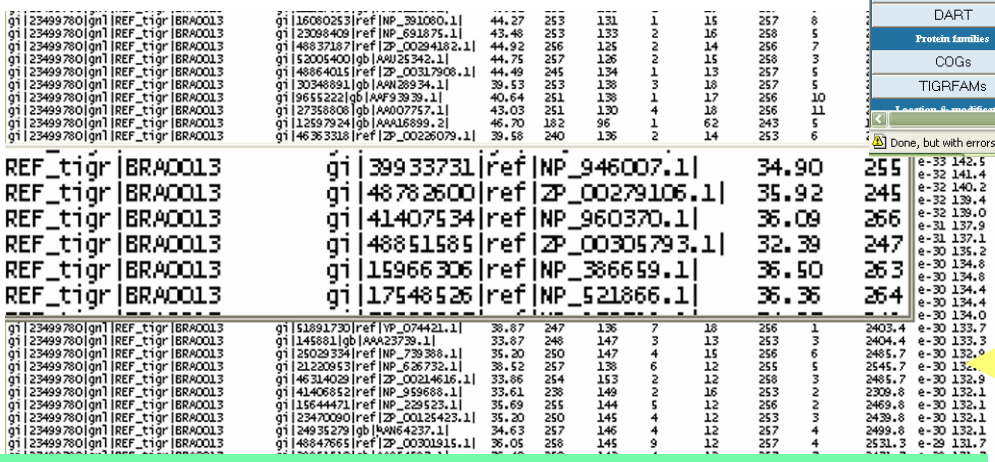


PUMA: Analysis of Metabolism

PUMA Knowledge Base

Information about proteins analyzed against ~2 million gene sequences





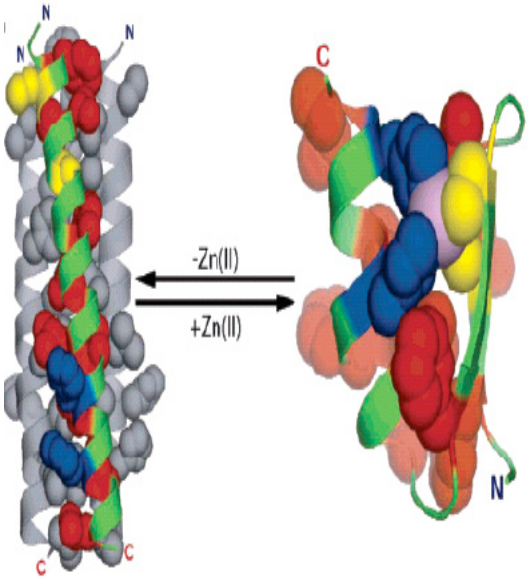
Analysis on Grid

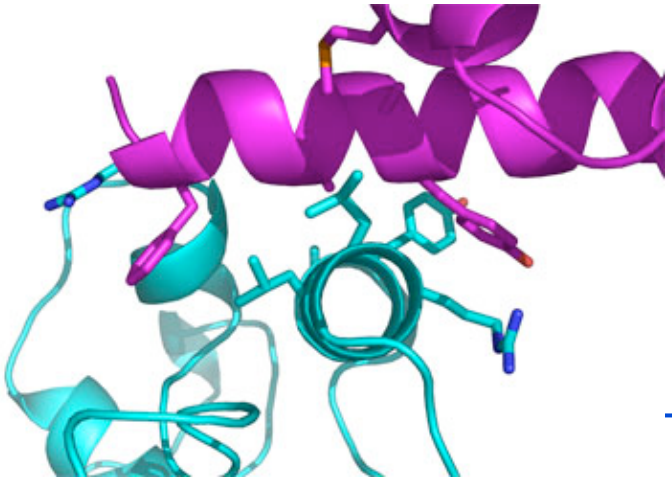
Involves millions of BLAST, BLOCKS, and other processes

Natalia Maltsev et al.
http://compbio.mcs.anl.gov/puma2

Sample Engagement: Kuhlman Lab

Using OSG to design proteins that adopt specific three dimensional structures and bind and regulate target proteins important in cell biology and pathogenesis. These designed proteins are used in experiments with living cells to detect when and where the target proteins are activated in the cells

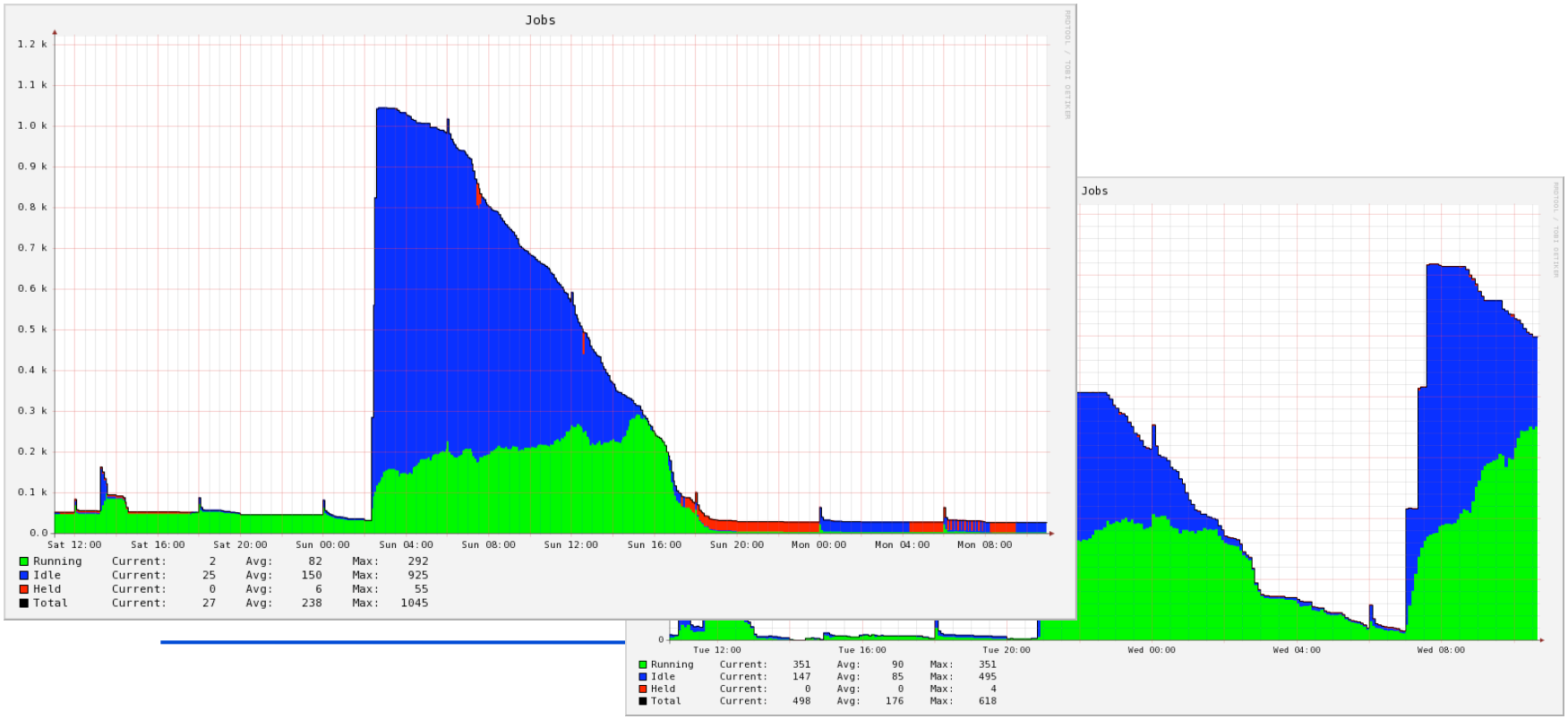




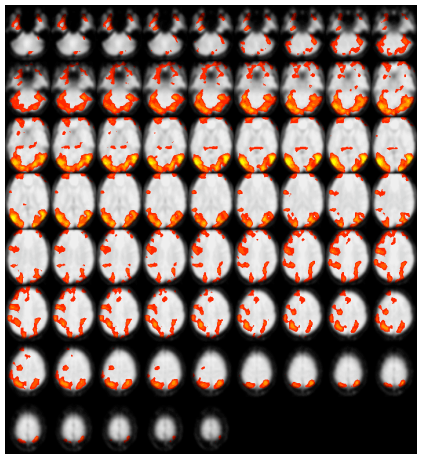
Sr. Rosetta Researcher and his team, little CS/IT expertise, no grid expertise. Quickly up and running with large scale jobs across OSG, **>250k CPU hours**

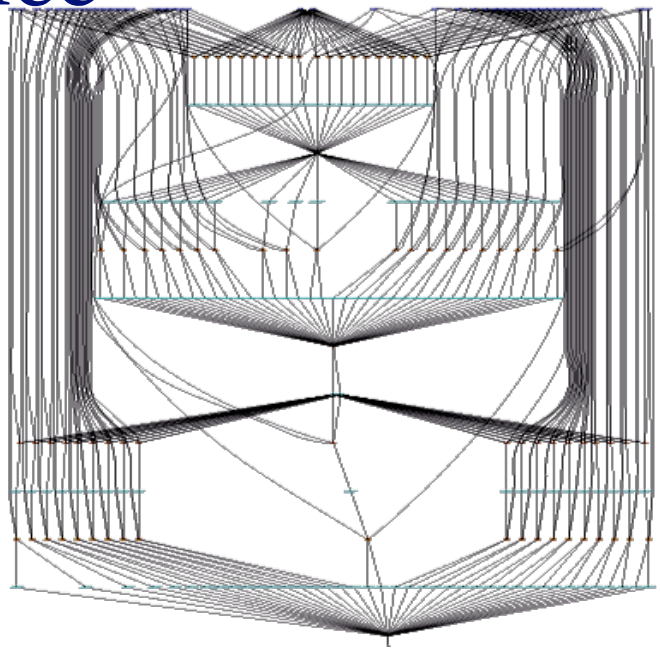
Rosetta on OSG

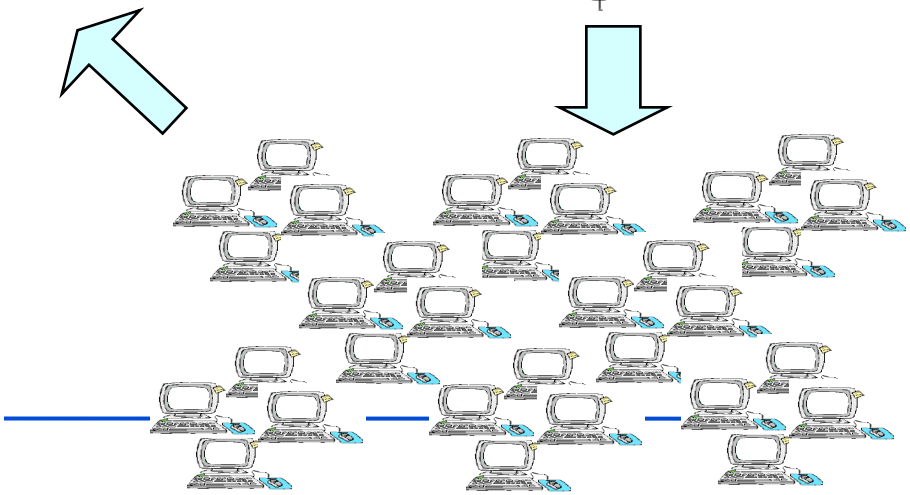
- Each protein design requires about 5,000 CPU hours, distributed across 1,000 individual compute jobs.



Workflow Motivation: example from Neuroscience

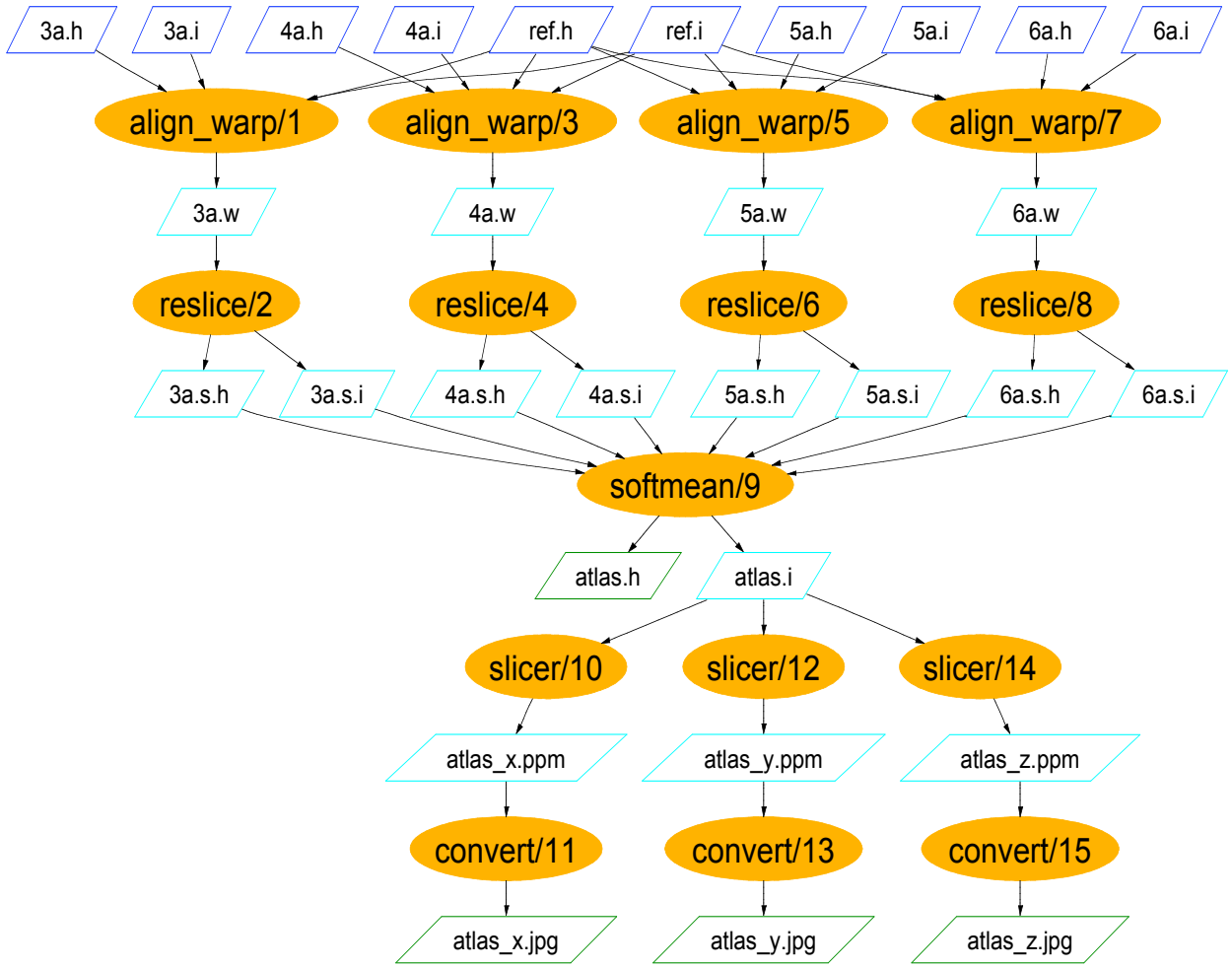


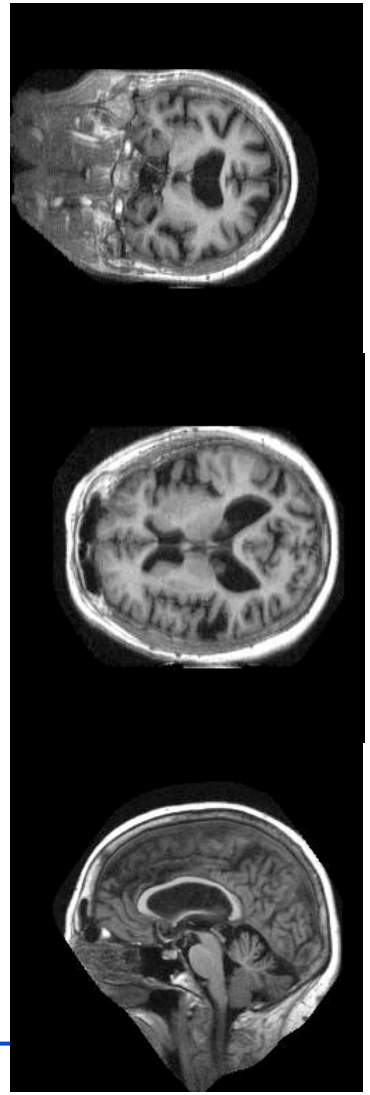




- Large fMRI datasets
  - 90,000 volumes / study
  - 100s of studies
- Wide range of analyses
  - Testing, production runs
  - Data mining
  - Ensemble, Parameter studies

A typical workflow pattern in fMRI image analysis runs many filtering apps.





Workflow courtesy James Dobson, Dartmouth Brain Imaging Center

Introduction to Grid Computing Tutorial Outline

I. Motivation and Grid Architecture
II. Grid Examples from Life Sciences
III. Grid Security
IV. Job Management: Running Applications
V. Data Management
VI. Open Science Grid and TeraGrid
VII. Workflow on the Grid
VIII. Next steps: Learning more, getting started

Grid security is a crucial component

- Problems being solved might be sensitive
- Resources are typically valuable
- Resources are 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

Grid Security Infrastructure - GSI

- Provides secure communications for all the higher-level grid services
- Secure *Authentication* and *Authorization*
  - Authentication ensures you *are* whom you claim to be
    - *ID card, fingerprint, passport, username/password*
  - Authorization controls what you are permitted to *do*
    - *Run a job, read or write a file*
- GSI provides Uniform Credentials
- Single Sign-on
  - User authenticates once – then can perform many tasks

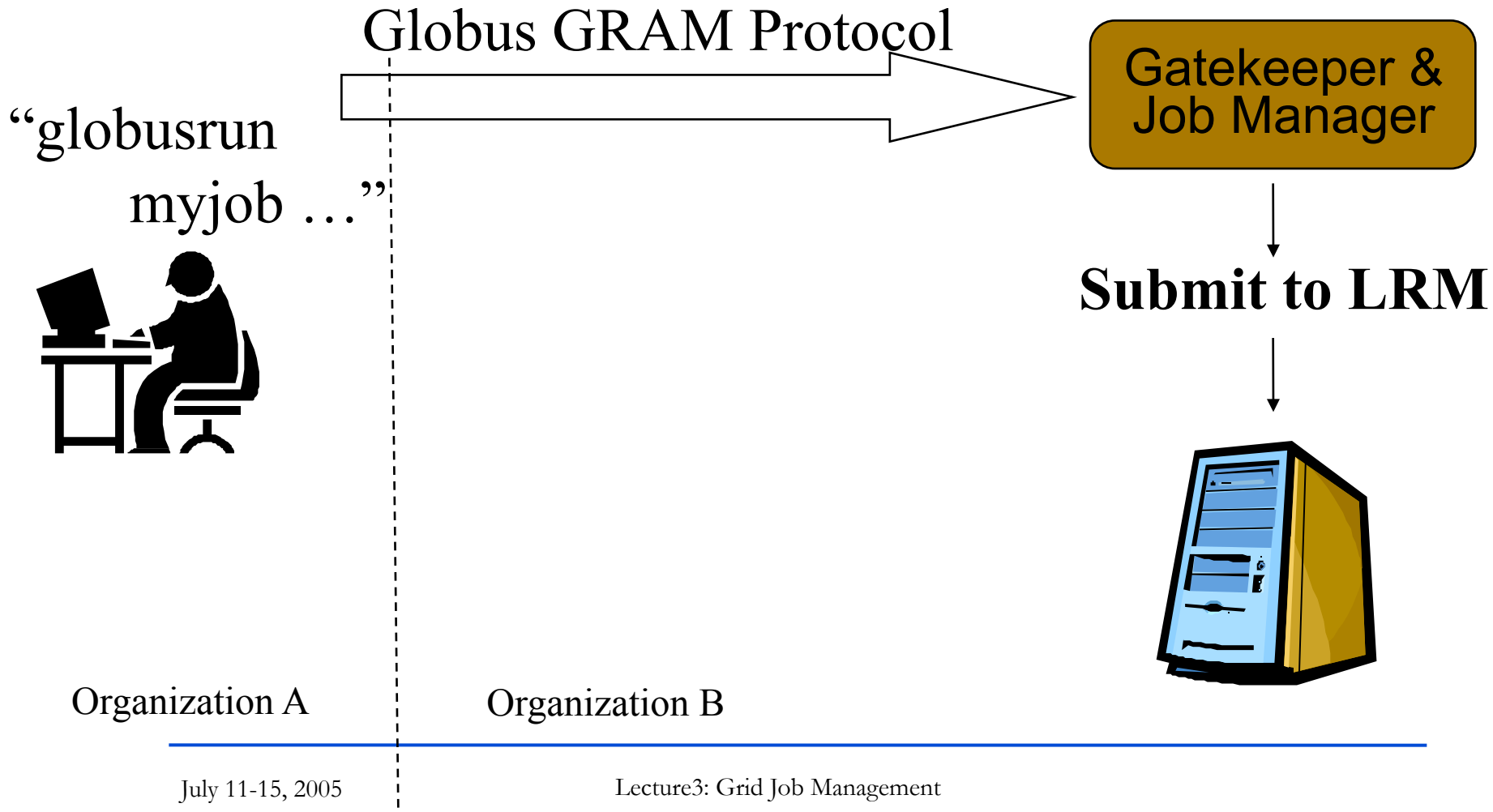
Introduction to Grid Computing Tutorial Outline

I. Motivation and Grid Architecture
II. Grid Examples from Life Sciences
III. Grid Security
IV. Job Management: Running Applications
V. Data Management
VI. Open Science Grid and TeraGrid
VII. Workflow on the Grid
VIII. Next steps: Learning more, getting started

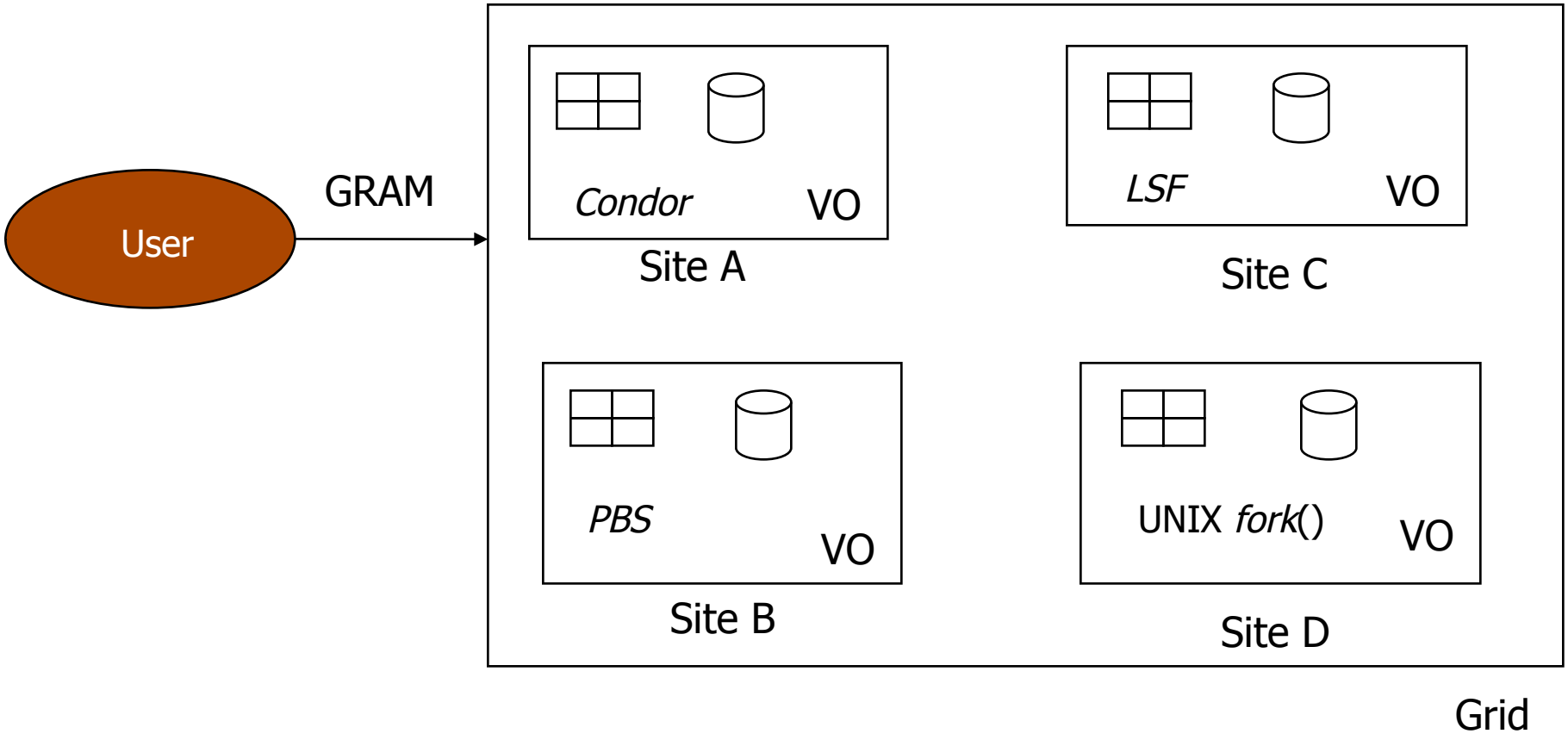
Local Resource Manager: a batch scheduler for running jobs on a computing cluster

- Popular LRMs include:
  - PBS – Portable Batch System
  - LSF – Load Sharing Facility
  - SGE – Sun Grid Engine
  - Condor – Originally for cycle scavenging, Condor has evolved into a comprehensive system for managing computing
- LRMs execute on the cluster's *head node*
- Simplest LRM allows you to "fork" jobs quickly
  - Runs on the head node (*gatekeeper)* for fast utility functions
  - No queuing (but this is emerging to "throttle" heavy loads)
- In GRAM, each LRM is handled with a "job manager"

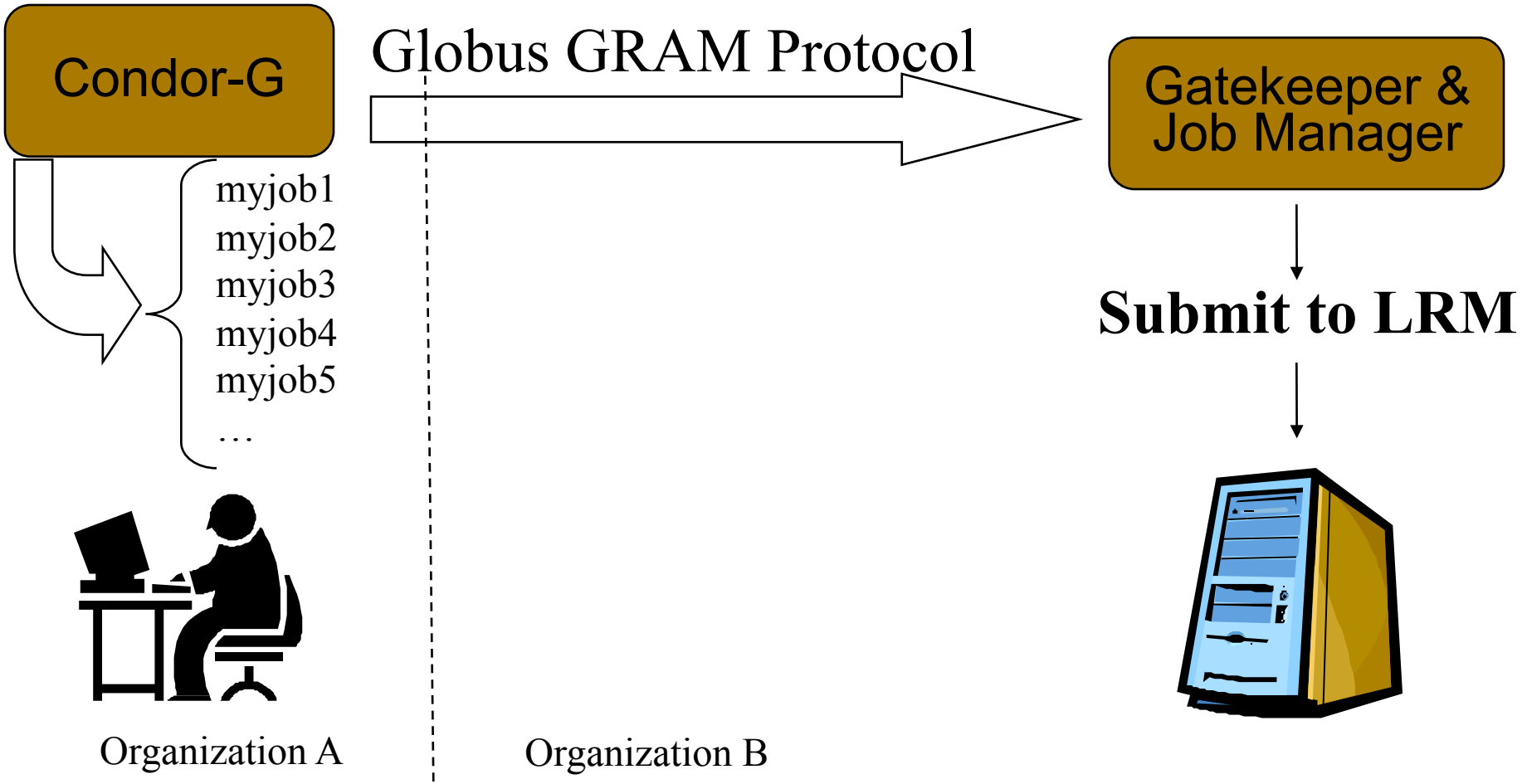
GRAM – Globus Resource Allocation Manager



GRAM provides a uniform interface to diverse cluster schedulers.



Condor-G: Grid Job Submission Manager



Introduction to Grid Computing Tutorial Outline

I. Motivation and Grid Architecture
II. Grid Examples from Life Sciences
III. Grid Security
IV. Job Management: Running Applications
V. Data Management
VI. Open Science Grid and TeraGrid
VII. Workflow on the Grid
VIII. Next steps: Learning more, getting started

Data management services provide the mechanisms to find, move and share data

- GridFTP
  - Fast, Flexible, Secure, Ubiquitous data transport
  - Often embedded in higher level services
- RFT
  - Reliable file transfer service using GridFTP
- Replica Location Service (RLS)
  - Tracks multiple copies of data for speed and reliability
  - Emerging services integrate replication and transport
- Metadata management services are evolving

GridFTP is secure, reliable and fast

- Security through GSI
  - Authentication and authorization
  - Can also provide encryption
- Reliability by restarting failed transfers
- Fast and scalable
  - Handles huge files (>4GB – 64bit sizes)
  - Can set TCP buffers for optimal performance
  - Parallel transfers
  - Striping (multiple endpoints)
- Client Tools

  globus-url-copy, uberftp, custom clients

GridFTP

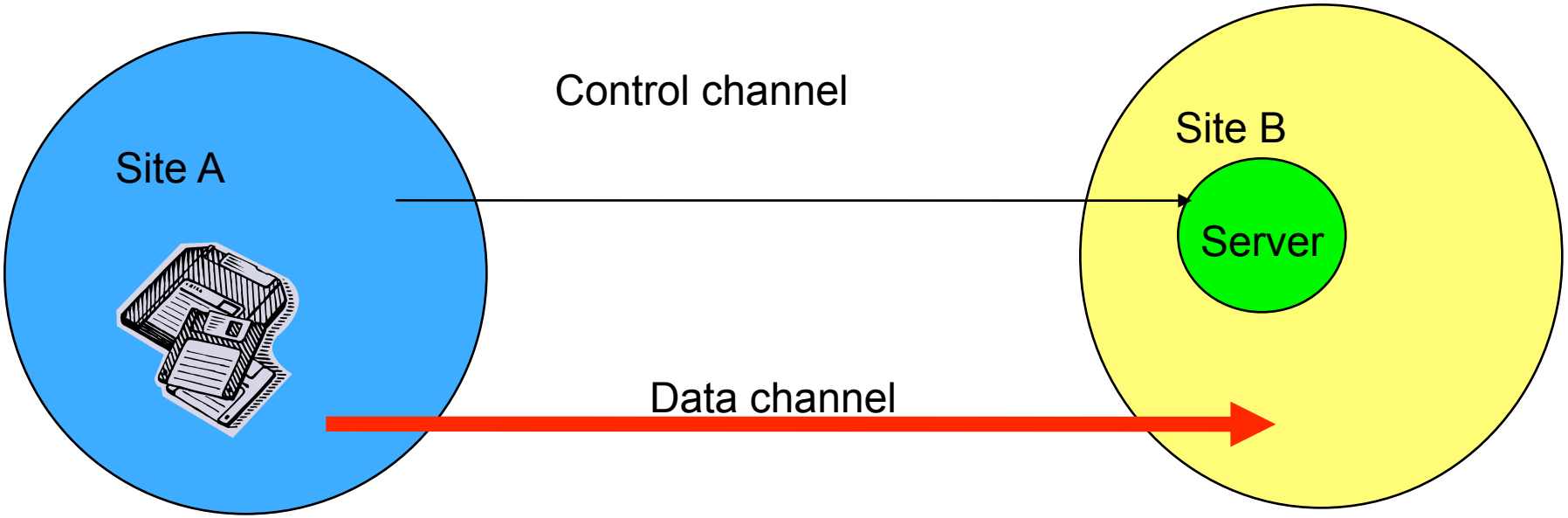
- Extensions to well known FTP File Transfer Protocol
  - Strong authentication, encryption via Globus GSI
  - Multiple, parallel data channels
  - Third-party transfers
  - Tunable network & I/O parameters
  - Server side processing, command pipelining

Basic Definitions

- Control Channel
  - TCP link over which commands and responses flow
  - Low bandwidth; encrypted and integrity protected by default
- Data Channel
  - Communication link(s) over which the actual data of interest flows
  - High Bandwidth; authenticated by default; encryption and integrity protection optional

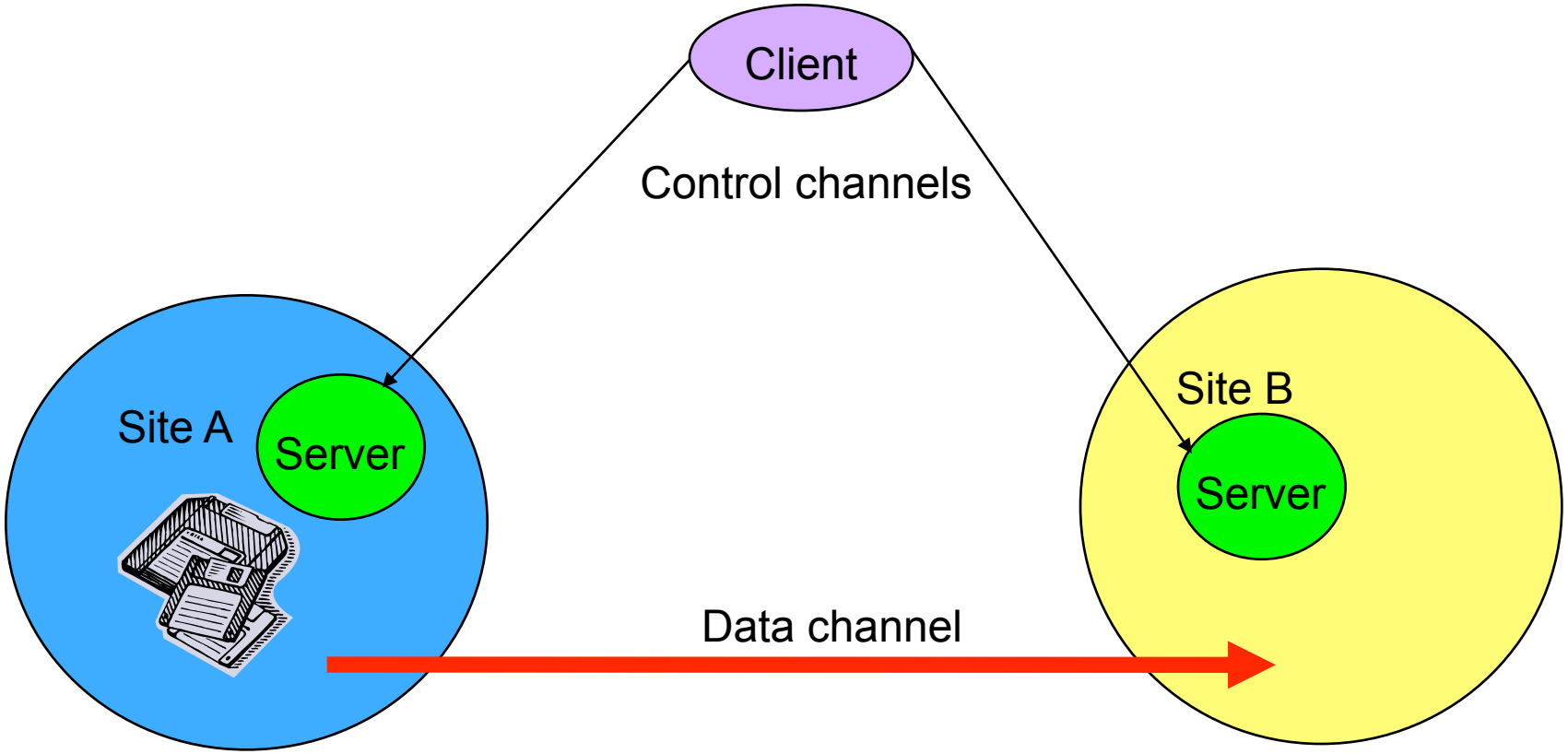
A file transfer with GridFTP

- Control channel can go either way
  - Depends on which end is client, which end is server
- Data channel is still in same direction



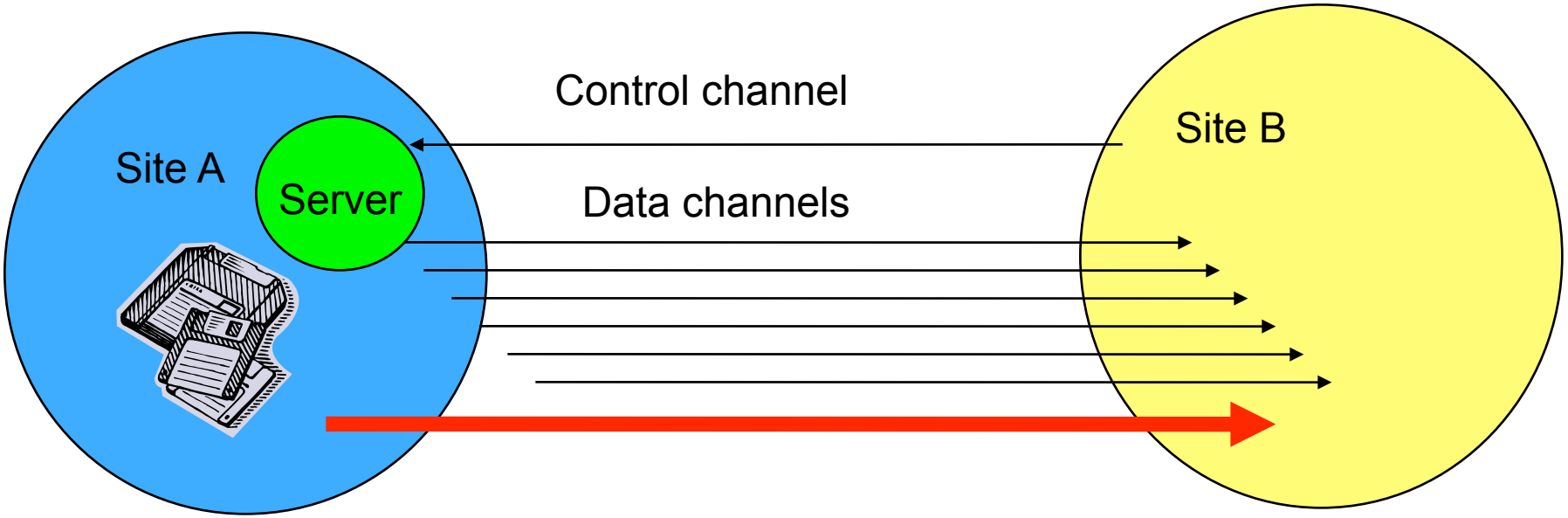
Third party transfer

- Controller can be separate from src/dest
- Useful for moving data from storage to compute



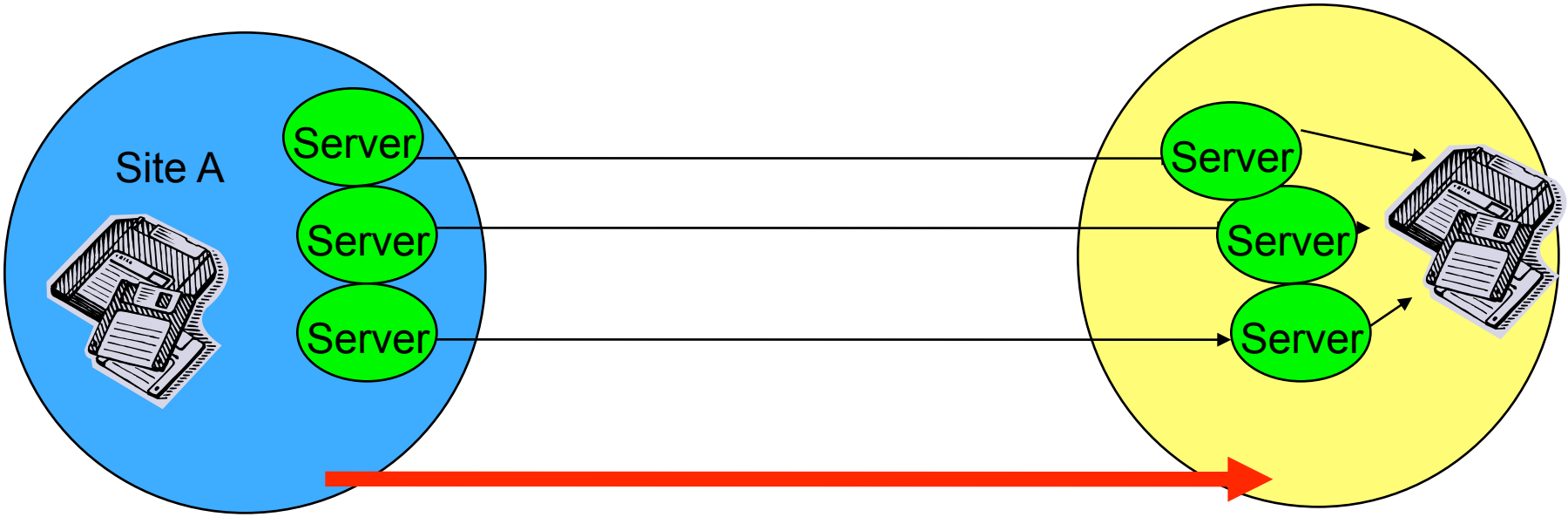
Going fast – parallel streams

- Use several data channels



Going fast – striped transfers

- Use several servers at each end
- Shared storage at each end



GridFTP examples

- **globus-url-copy**

  file:///home/YOURLOGIN/datatex/myfile

  gsiftp://osg-edu.cs.wisc.edu/nfs/osgedu/YOURLOGIN/ex1

- **globus-url-copy**

gsiftp://osg-edu.cs.wisc.edu/nfs/osgedu/YOURLOGIN/ex2

gsiftp://tp-osg.ci.uchicago.edu/YOURLOGIN/ex3

Grids replicate data files for faster access

- Effective use of the grid resources – more parallelism
- Each *logical* file can have multiple *physical* copies
- Avoids single points of failure
- Manual or automatic replication
  - Automatic replication considers the demand for a file, transfer bandwidth, etc.

File catalogues tell you where the data is

- File Catalog Services
  - Replica Location Service (RLS)
  - Phedex
  - RefDB / PupDB
- Requirements
  - Abstract the logical file name (LFN) for a physical file
  - maintain the mappings between the LFNs and the PFNs *(physical file names)*
  - Maintain the location information of a file

RLS -Replica Location Service

- RLS maps logical filenames to physical filenames
- Logical Filenames (LFN)
  - A symbolic name for a file – form is up to application
  - Does not refer to file location (host, or where on host)
- Physical Filenames (PFN)
  - Refers to a file on some filesystem somewhere
  - Often use `gsiftp://` URLs to specify PFNs
- Two RLS catalogs:
  - Local Replica Catalog (LRC) and
  - Replica Location Index (RLI)

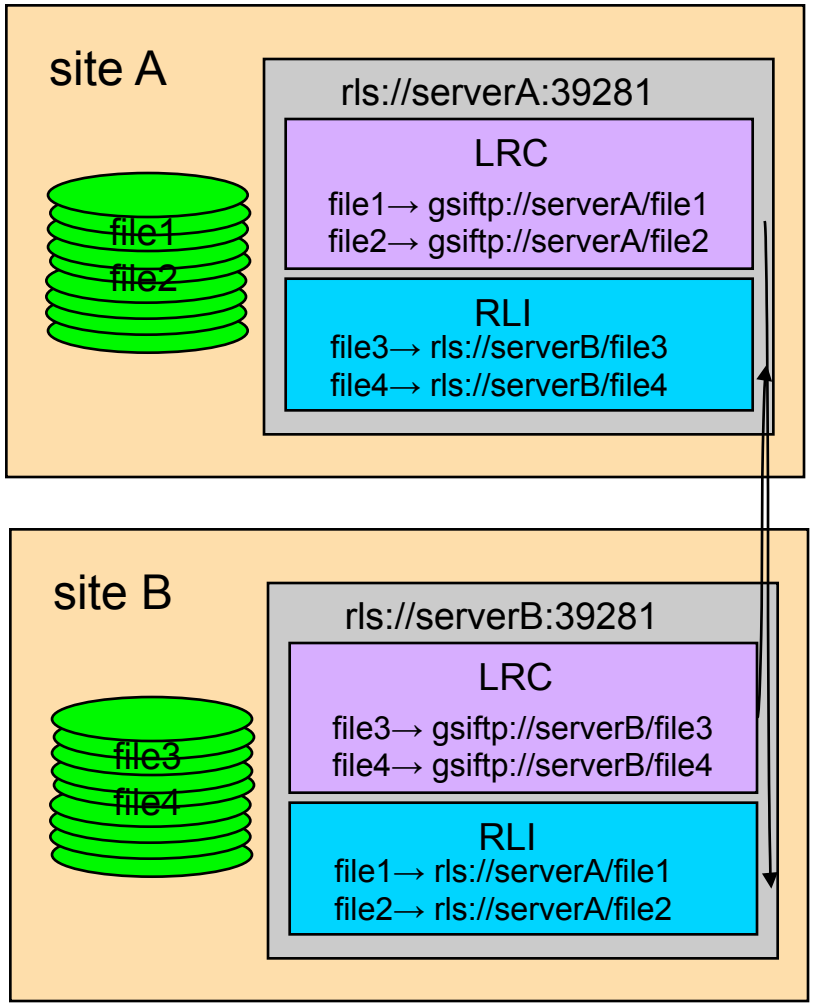
Local Replica Catalog (LRC)

- stores mappings from LFNs to PFNs.
- Interaction:
  - *Q:* Where can I get filename ‘experiment_result_1’?
  - *A:* You can get it from gsiftp://gridlab1.ci.uchicago.edu/home/benc/r.txt
- Undesirable to have one of these for whole grid
  - Lots of data
  - Single point of failure

Replica Location Index (RLI)

- stores mappings from LFNs to LRCs.
- Interaction:
  - Q: where can 1 find filename ‘experiment_result_1
  - *A:* You can get more info from the LRC at gridlab1
  - (Then go to ask that LRC for more info)
- Failure of an RLI or RLC doesn’t break RLS
- RLI stores reduced set of information
  - can handle many more mappings

Globus RLS


site A
rls://serverA:39281
LRC
file1→ gsiftp://serverA/file1
file2→ gsiftp://serverA/file2
RLI
file3→ rls://serverB/file3
file4→ rls://serverB/file4
file1
file2
site B
rls://serverB:39281
LRC
file3→ gsiftp://serverB/file3
file4→ gsiftp://serverB/file4
RLI
file1→ rls://serverA/file1
file2→ rls://serverA/file2
file3
file4

The LIGO Data Grid integrates GridFTP & RLS

LIGO Gravitational Wave Observatory

Replicating >1 Terabyte/day to 8 sites

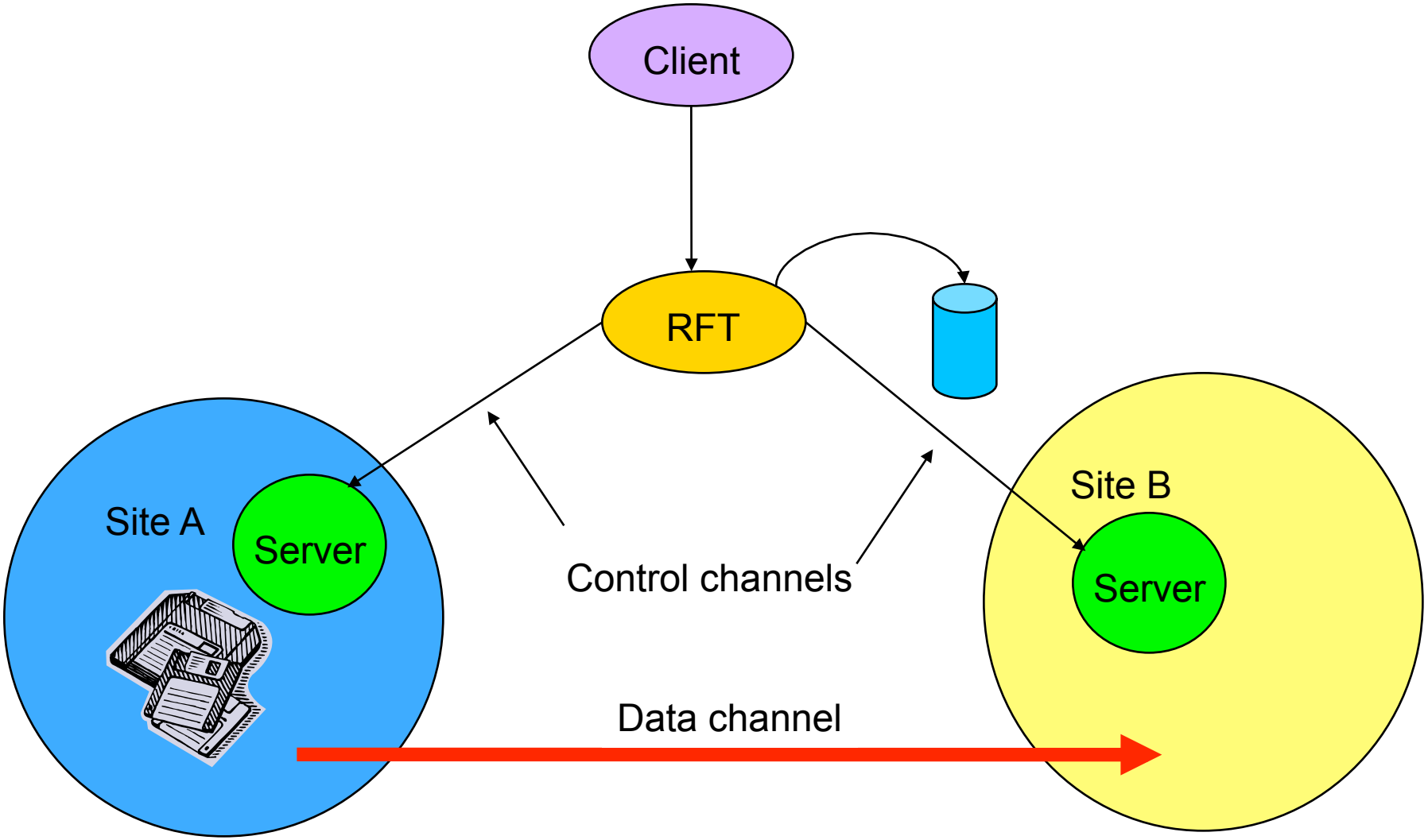
>40 million replicas so far

MTBF = 1 month

RFT - Reliable File Transfer

- Provides recovery for additional failure possibilities beyond those covered by GridFTP
  - Uses a database to track transfer requests and their status
  - Users start transfers – then RFT takes over
  - Integrated Automatic Failure Recovery.
    - Network level failures.
    - System level failures etc.
    - Utilizes restart markers generated by GridFTP

Reliable file transfer



What is SRM?

- Storage Resource Managers (SRMs) are middleware components
  - whose function is to provide
    - dynamic space allocation
    - file management

    on shared storage resources on the Grid
  - Different implementations for underlying storage systems are based on the same SRM specification

SRMs role in grid

- SRMs role in the data grid architecture
  - Shared storage space allocation & reservation
    - important for data intensive applications
  - Get/put files from/into spaces
    - archived files on mass storage systems
  - File transfers from/to remote sites, file replication
  - Negotiate transfer protocols
  - File and space management with lifetime
  - support non-blocking (asynchronous) requests
  - Directory management
  - Interoperate with other SRMs

SRM: Main concepts

- Space reservations
- Dynamic space management
- Pinning file in spaces
- Support abstract concept of a file name: Site URL
- Temporary assignment of file names for transfer: Transfer URL
- Directory management and authorization
- Transfer protocol negotiation
- Support for peer to peer request
- Support for asynchronous multi-file requests
- Support abort, suspend, and resume operations
- Non-interference with local policies