Enabling Science and Engineering Applications on Grids

Ed Seidel

Center for Computation & Technology

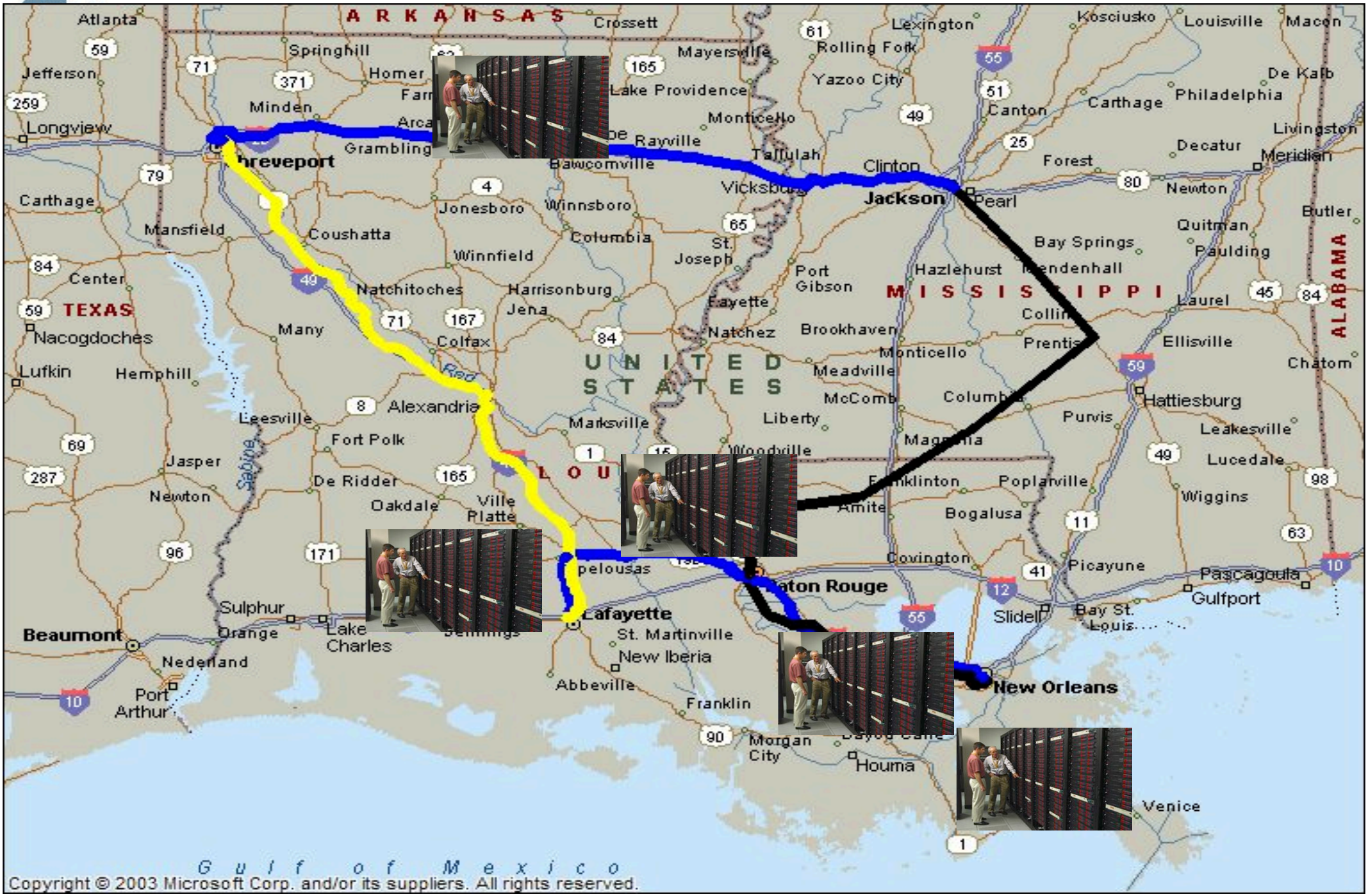
Louisiana State University

Albert-Einstein-Institut

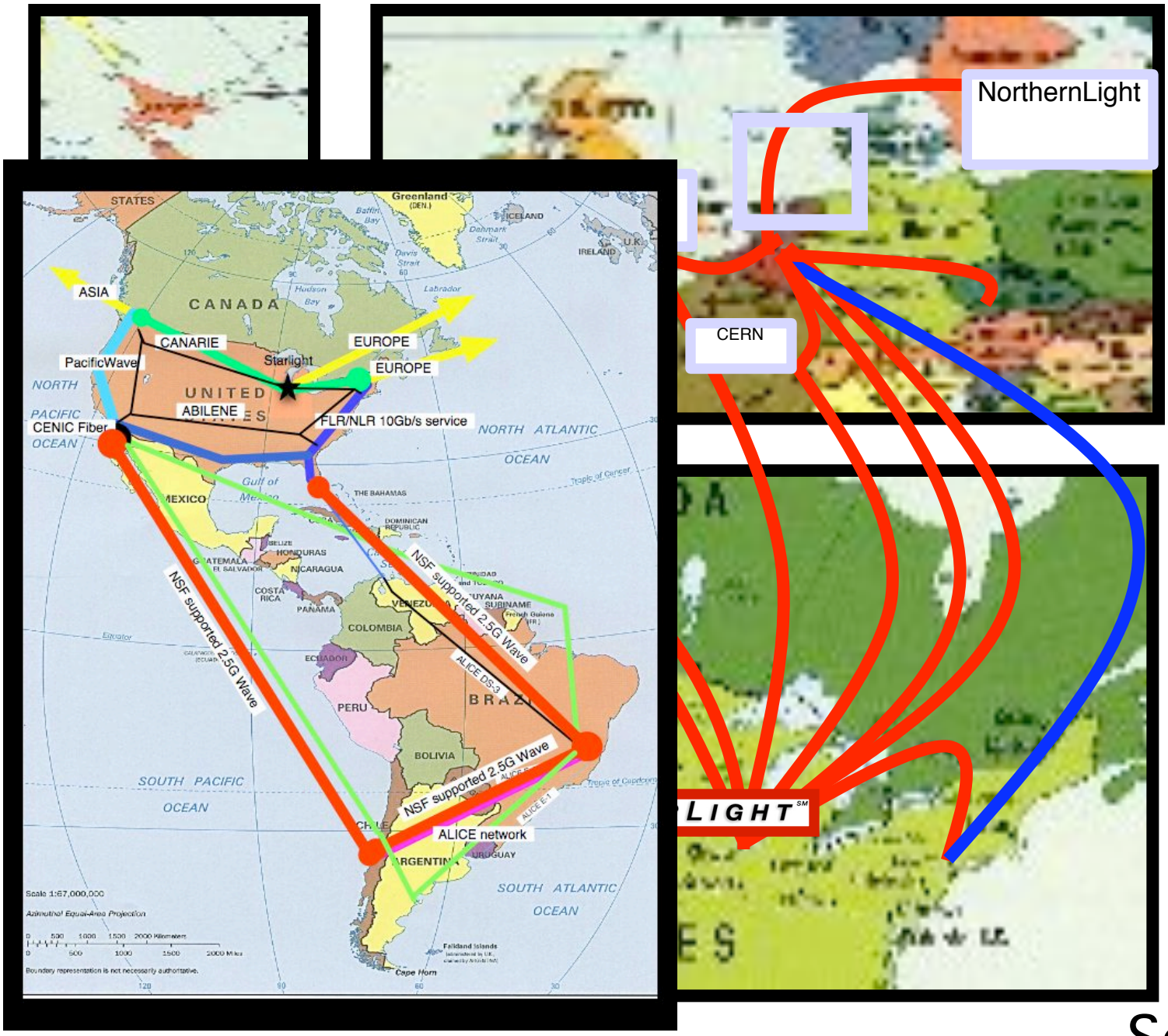
Potsdam, Germany

Optical Networks

- New technology for decade: 1000x faster than typical regional networks: Louisiana statewide network (LONI)
- National Lambda Rail
  - $100M USA Optical Network
  - Backbone for next gen. research
  - Locally funded!! No Federal $$
- Other countries: $100'sM
  - Canada, Poland, Holland, Czech, etc; linked!
- 2 dozen+ states committed
- Southern US states investing to be competitive for federal funding, industry


ARKANSAS
TEXAS
MISSISSIPPI
ALABAMA
UNITED STATES
Gulf of Mexico
Shreveport
Jackson
Lafayette
New Orleans
Baton Rouge
Lake Charles
Beaumont
Port Arthur
Hattiesburg
Meridian
Natchez
Vicksburg
Alexandria
Monroe
Texarkana
Longview
Nacogdoches
Lufkin
Houma
Slidell
Gulfport
Pascagoula
Biloxi
Picayune
Venice
Copyright © 2003 Microsoft Corp. and/or its suppliers. All rights reserved.

2004 International Lambdas



European lambdas to US
–10Gb Amsterdam—Chicago
–10Gb London—Chicago
–3Gb CERN — Chicago

Canadian lambdas to US
–10Gb Chicago-Canada-NYC
–10Gb Chicago-Canada-Seattle

US lambda to Europe
–7Gb Chicago—Amsterdam

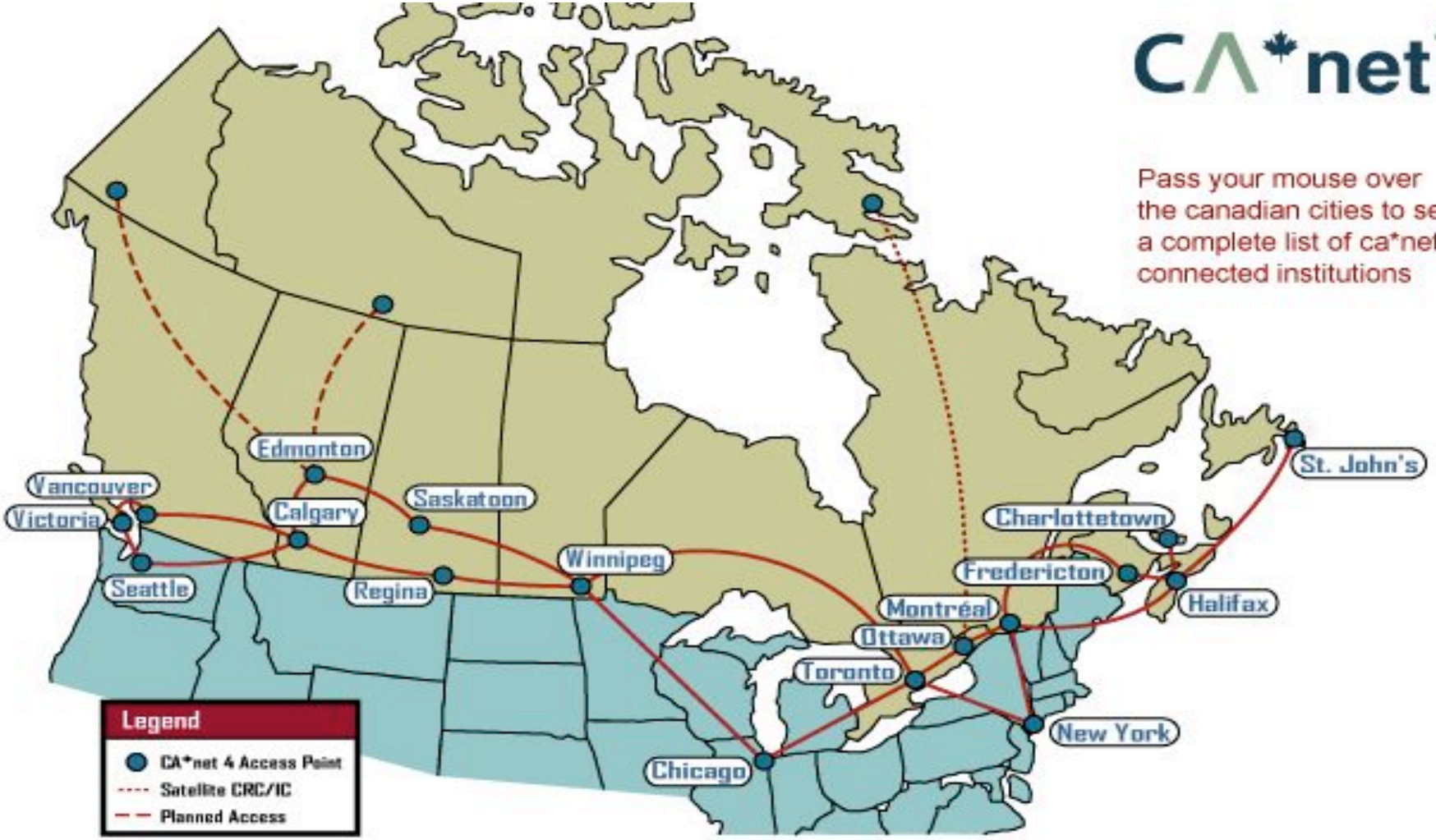
US/Japan lambda
–10Gb Chicago—Tokyo

European lambdas
–10Gb Amsterdam—CERN
–2.5Gb Prague—Amsterdam
–2.5Gb Stockholm—Amsterdam
–10Gb London—Amsterdam

IEEAF lambdas (blue)
–10Gb NYC—Amsterdam
–10Gb Seattle—Tokyo

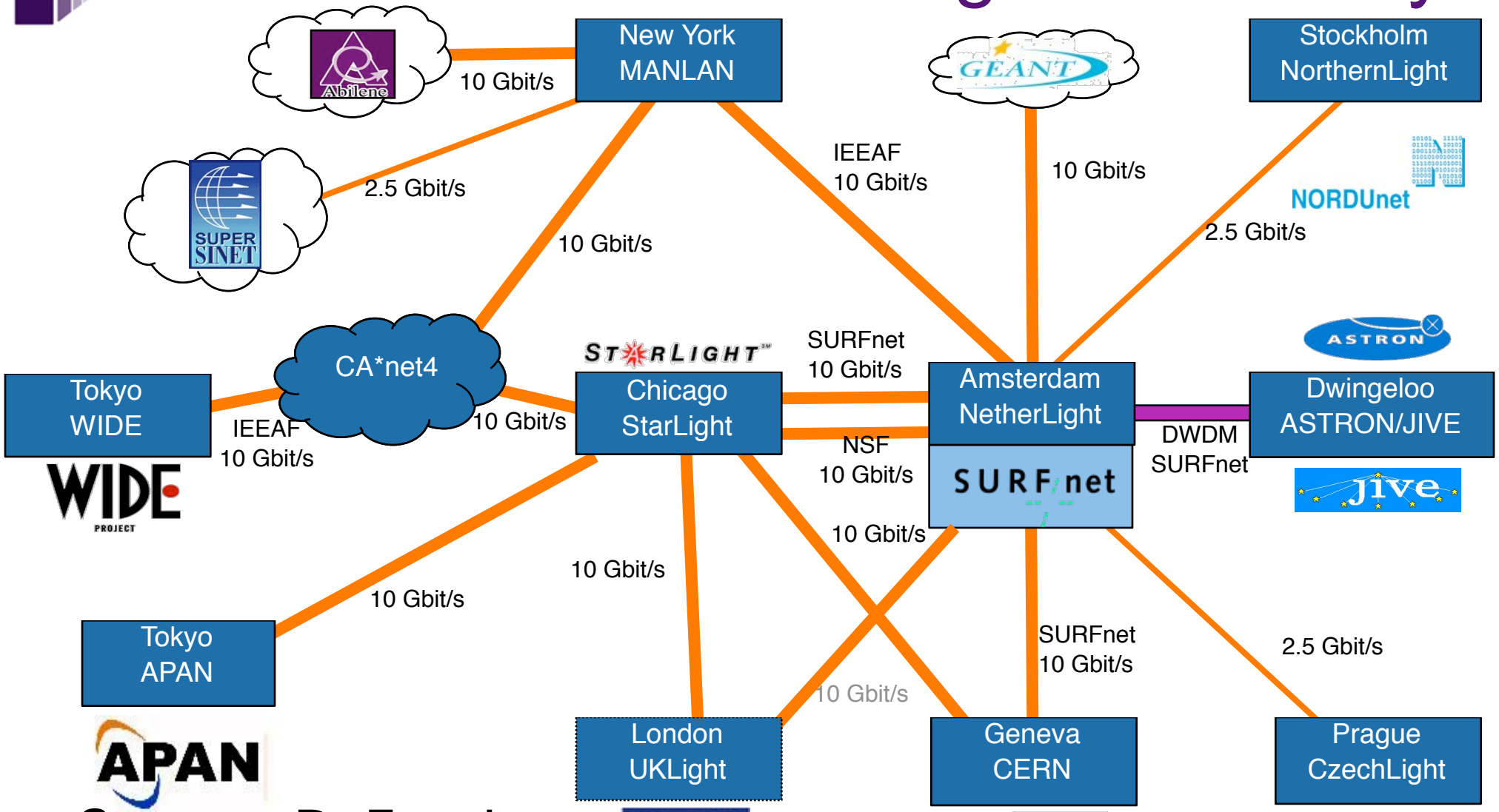
Source: DeFanti

CA*net4 has 2x10Gb Lambdas, More Coming



Source: DeFanti

Global Lambda Integrated Facility



Source: DeFanti

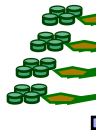
Source: Kees Neggers, SURFnet

What do we want to do with this?

- Collaboration
  - Distributed communities share resources: GWEN, GEON, NEES, etc.
  - “Shared Cyberinfrastructure”: data, code, tools, resources, simulations…
- “Standard Things”
  - Task Farming, Resource Brokering, Remote Steering, Managing Data (!)
- “New Scenarios”
  - Apps abstracted, become services
  - Dynamic apps decide their future, find their services (data, resources, applications, people)
  - Apps distributed, spawned, task farmed, controlled, monitored with other apps/people

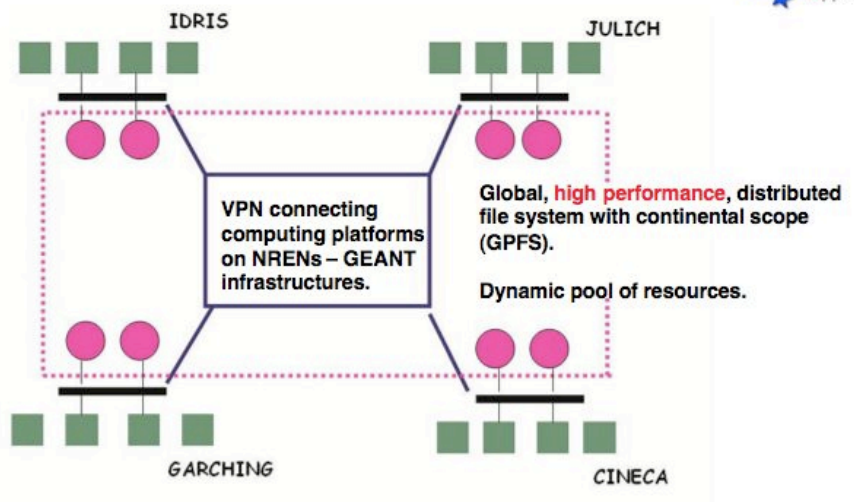
Grids: Bringing all Together

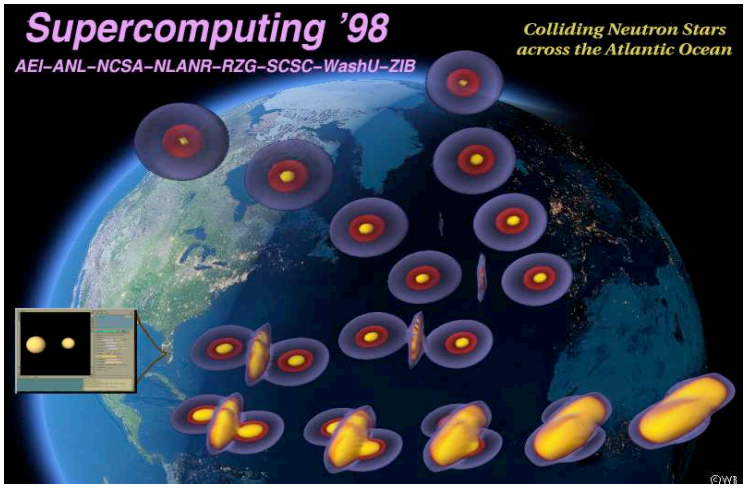
- Computational Devices Scattered Across the World
  - Compute servers (double 18 months): playstations, phones...
  - Networks (double each 9 months)
  - Sensors (exploding field)



- How to take advantage of this for science, engineering, business, art?
  - Harness multiple sites and devices
  - Deploy cyberinfrastructure: Globus, Unicore, many others

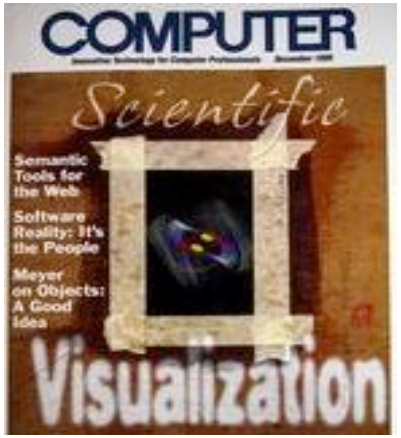
- Worries in US!
  - DOE, NSF cyberinfrastructure funding





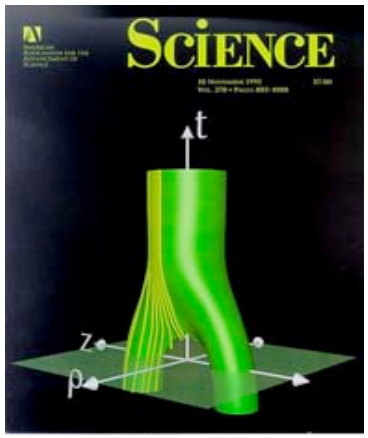
Gordon Bell Prize, 2001

Collaborations for Complex Problems



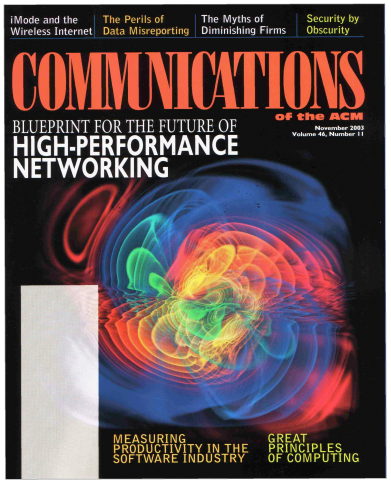
NASA Neutron Star Grand Challenge

- 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

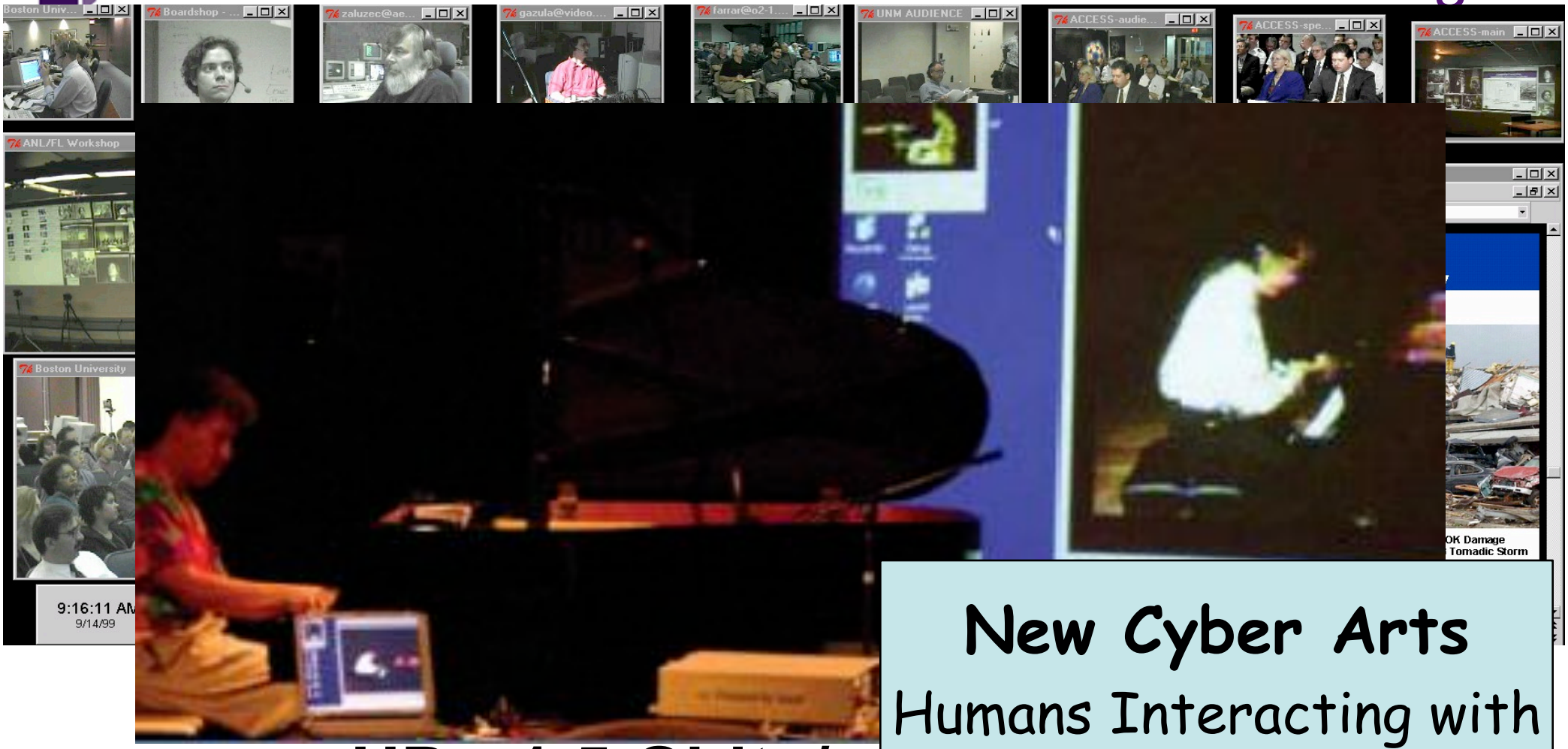
Examples of Future of Science & Engineering

- Require large scale data, simulations, beyond reach of any machine
- Require large geo-distributed cross-disciplinary collaborations
- Require Grid technologies

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
  - DDDAS
- Process/Simulation Driven
  - Demanding Simulations of Science and Engineering
  - Generators of PBytes of data

From Telephone Conference Calls to Access Grid International Video Meetings



New Cyber Arts

Humans Interacting with Virtual Realities

HD: 1.5 Gbits/s

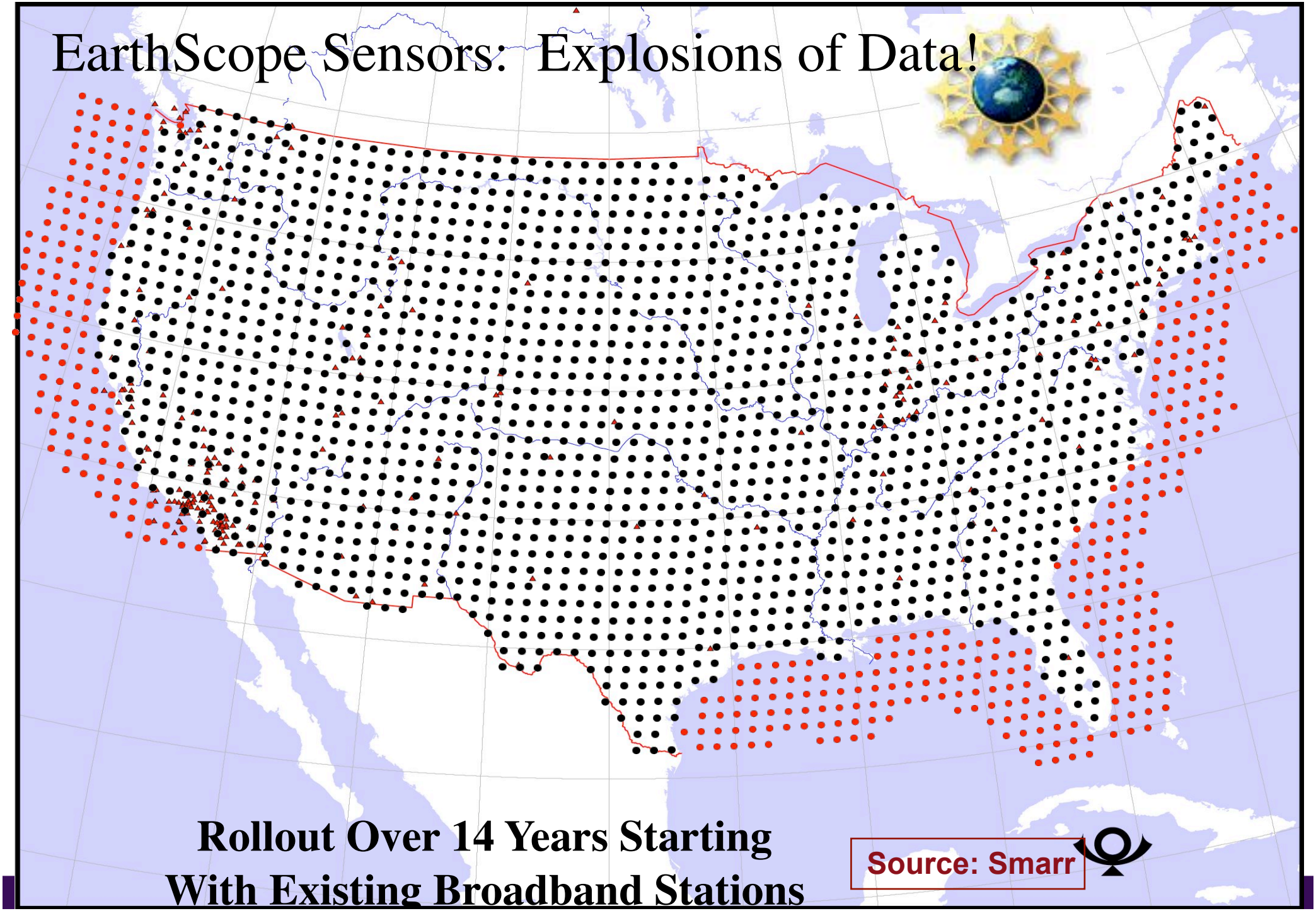
Internet Linked Pianos

Access Grid Lead-

NSF STARTAP Lead-UIC's Elec. Vis. Lab

Source: Smarr

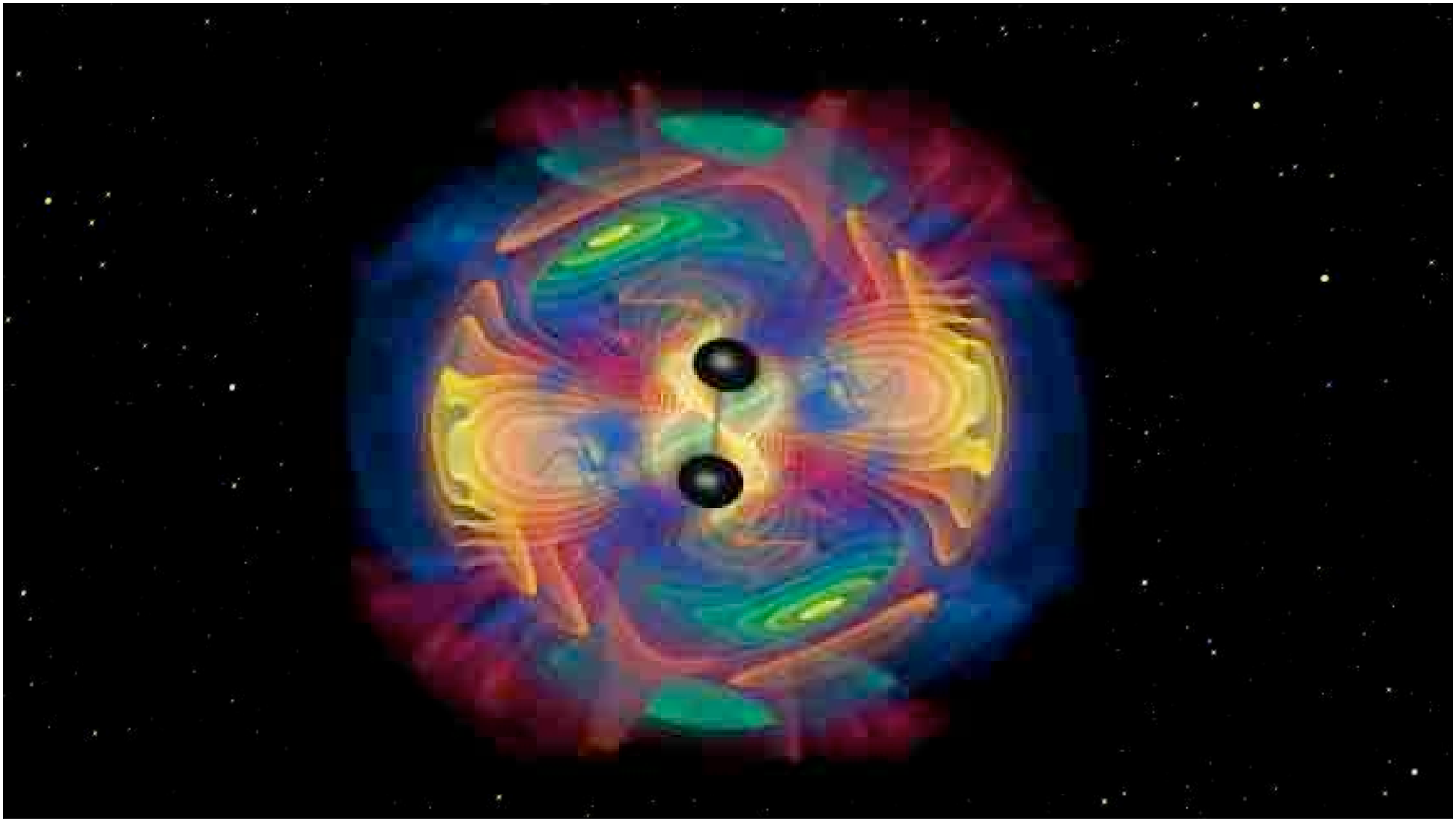
EarthScope Sensors: Explosions of Data!



Rollout Over 14 Years Starting With Existing Broadband Stations

Source: Smarr

Huge Black Hole Collision Simulation



Issues for Complex Simulations

- Huge amounts of data needed/generated across different sites
  - Smarr: supercomputers are merely petabyte generators
  - How to retrieve, track, manage data across Grid?
  - In this case, had to fly Berlin to NCSA, bring data back on disks!
- Many components developed by distributed collaborations
  - How to bring communities together?
  - How to find/load/execute different components?
- Many computational resources available
  - How to find best ones to start?
  - How to distribute work effectively?
- Needs of computations *change* with time!
  - How to adapt to changes?
  - How to monitor system?
- How to interact with experiments? Coming!
- Complex infrastructure accessed through common toolkits
  - Everyone should not reinvent the wheel (especially physicists)

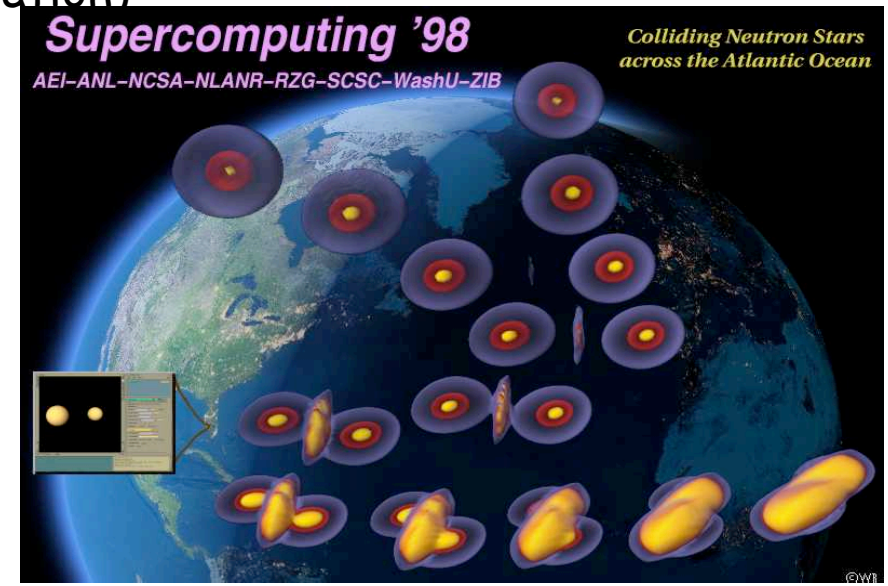
Computers as Petabyte Generators

(Black holes and more…)

- Crude vacuum of BH collisions today
  - 10^{15} Flops, Tbytes output, vastly downsampled!
- Desired BH, NS scenarios in few years
  - 10^{20} Flops, multiple orbits, adaptive meshes, hydro, 1 day on 50TF machine, 25TB+ output per run
  - May be physically distributed due to grid activities
- Real time scheduling across multiple resources for distributed computing
  - Lambda provisioning on demand: spawning (analysis, steering), migration, interactive viz from distributed collaborations
- Parameter Space! 10^3 - 10^6 simulations!

Distributed Computation: "Old" ways ofHarnessing Multiple Computers

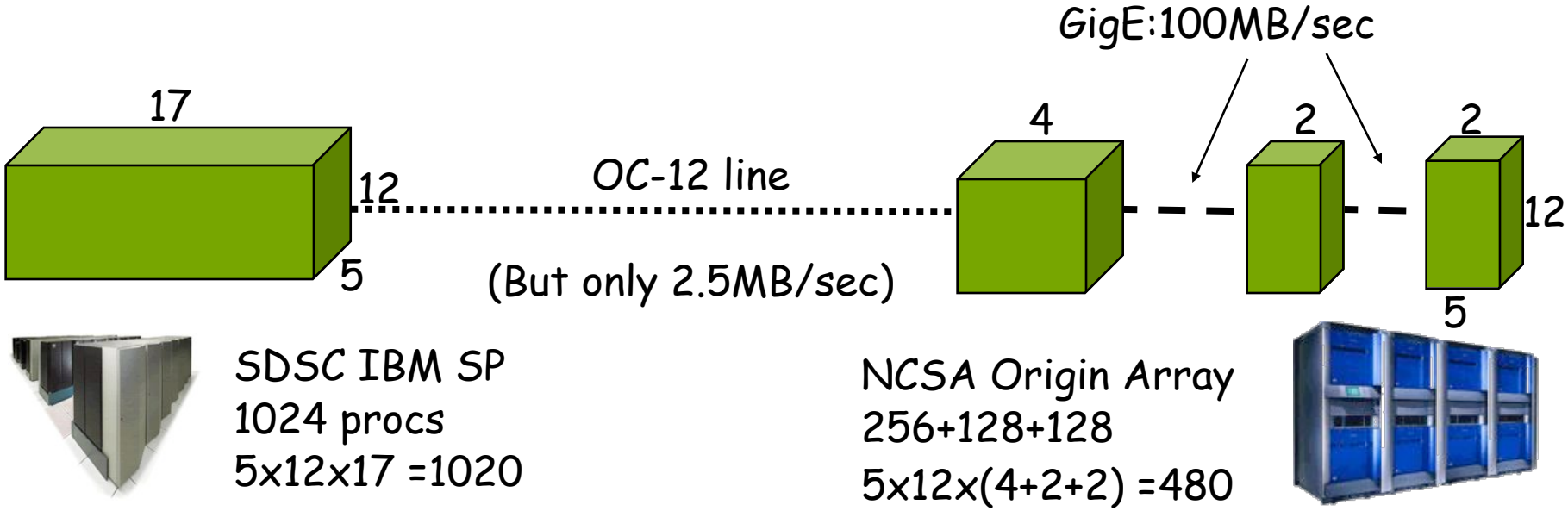
- Why would anyone want to do this?
  - Capacity: computers can't keep up with needs
  - Throughput
- Issues
  - Bandwidth (increasing faster than computation)
  - Latency
  - Communication needs, Topology
  - Communication/computation
- Techniques to be developed
  - Overlapping communication/computation
  - Extra ghost zones to reduce latency
  - Compression
  - Algorithms to do this for scientist



Dynamic Adaptive Distributed Computation

(T.Dramlitsch, with Argonne/U.Chicago)

GridLab



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, …

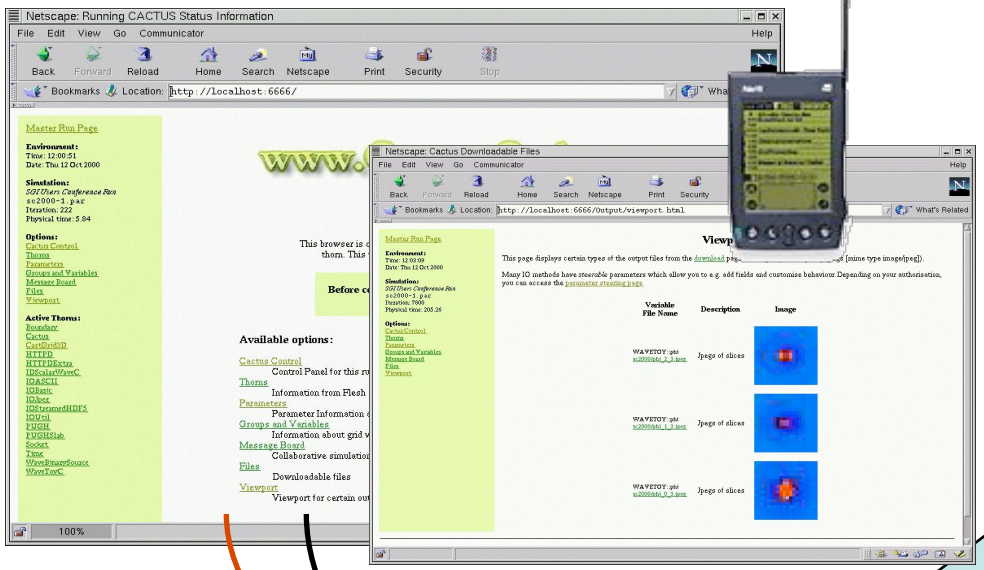
Won

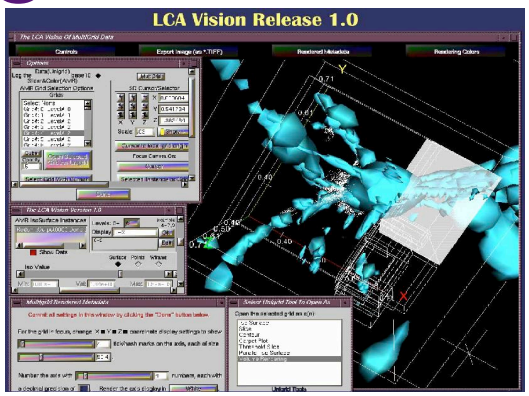
"Gordon Bell Prize" (Supercomputing 2001, Denver)

Cactus Framework: Enabling User Communities in Advanced HPC
www.cactuscode.org

- Toolkit for HPC, Grid, Collaborative Applications
- Abstract interfaces for everything
  - Parallelism, I/O, AMR, elliptic solvers, etc
  - Other packages, toolkits (Grace, Petsc, Samrai, HDF, Carpet, etc)
- Advanced capabilities
  - Streaming data, communication, Viz
- Grid capabilities
  - Globus, GAT, Portals, etc
- International developer/user base
  - We have openings for developers!!

Today: Interacting with Jobs





Any Viz Client:

LCA Vision, OpenDX

Changing any steerable parameter

- Parameters
- Physics, algorithms
- Performance

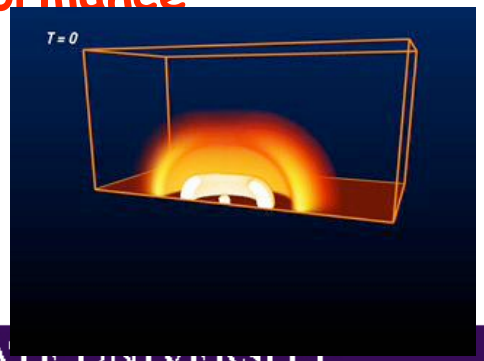
Remote Viz data

Streaming HDF5

Autodownsample

Remote Viz data

Amira

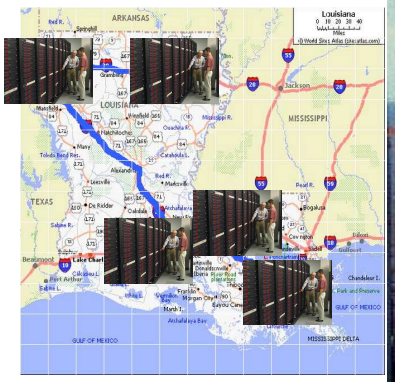


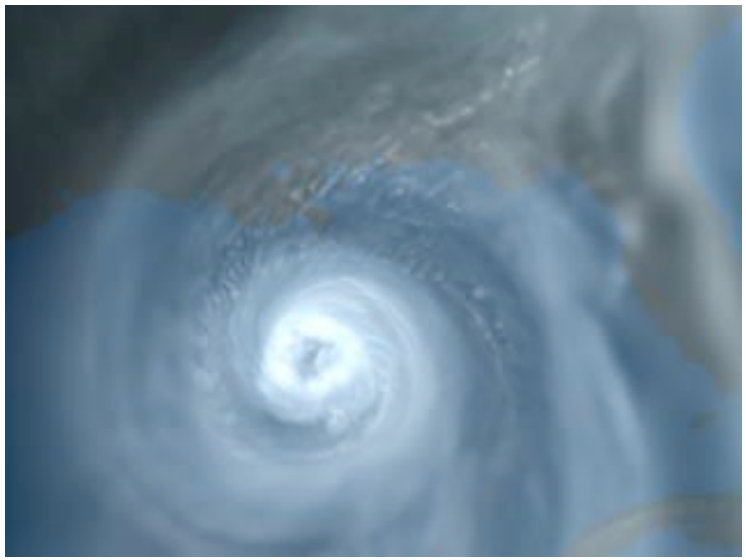
New Grid Applications: be creative

- Intelligent Parameter Surveys, Monte Carlos
  - May control other simulations!
- Dynamic Staging: move to faster/cheaper/bigger machine ("Grid Worm")
  - Need more memory? Need less?
- Multiple Universe: clone to investigate steered parameter ("Gird Virus")
- Automatic Component Loading
  - Needs of process change, discover/load/execute new component somewhere
- Automatic "Look Ahead:, convergence testing
  - spawn off and run coarser resolution to predict likely future, study convergence
- 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
- DDDAS: injecting data into the above, feed back to experiment

DDDAS Scenarios

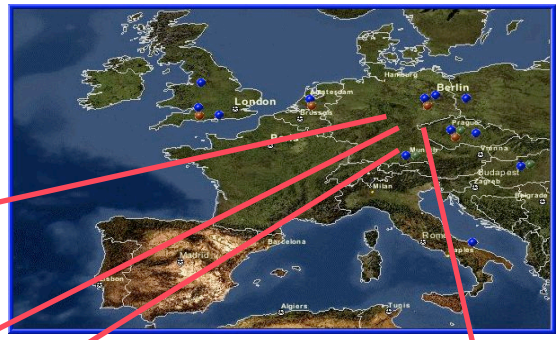
- UCoMS (Petroleum Engineering)
  - Deploy sensor networks across Gulf
  - Data collected to provide input to simulations, tasks farmed out
  - Results collected (transmitted back)
  - http://www.ucoms.org
- SCOOP (Ocean Observing)
  - Data coming in from sensors all over Gulf Realtime Operational Grid
  - Feeds in to models on Grid sites

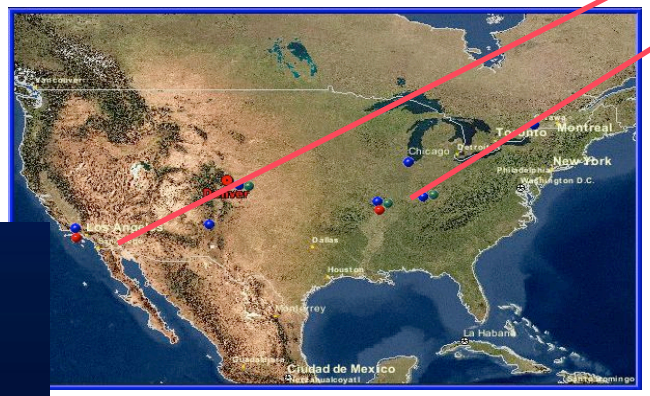


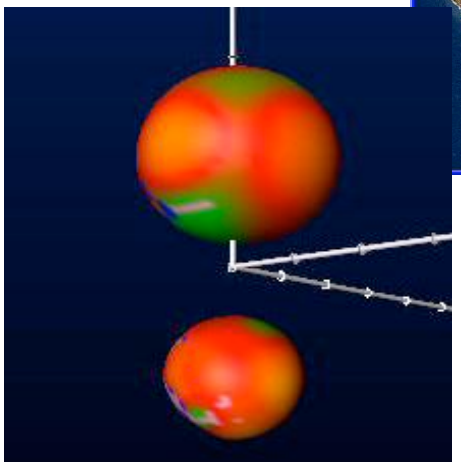


Spawning on ARG Testbed

Main Cactus BH Simulation starts here





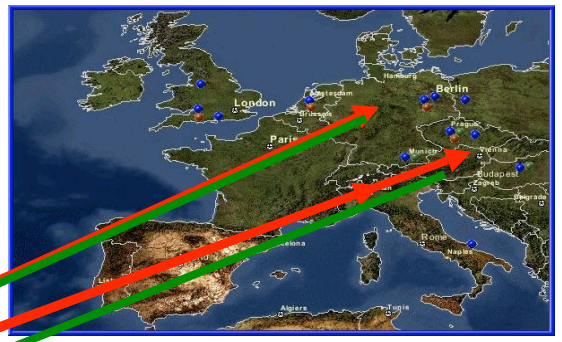


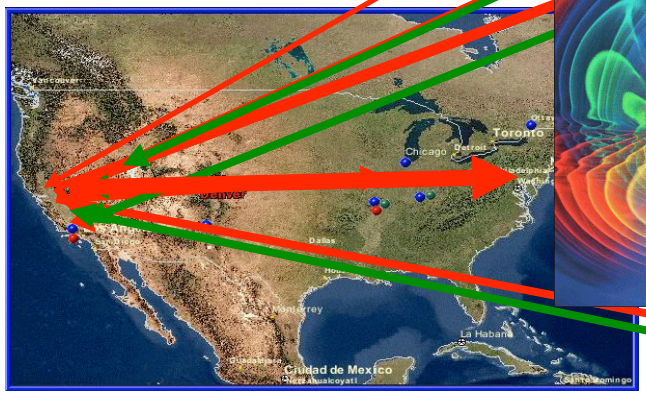
User only has to invoke "Spawner" thorn…

All analysis tasks spawned automatically to free resources worldwide

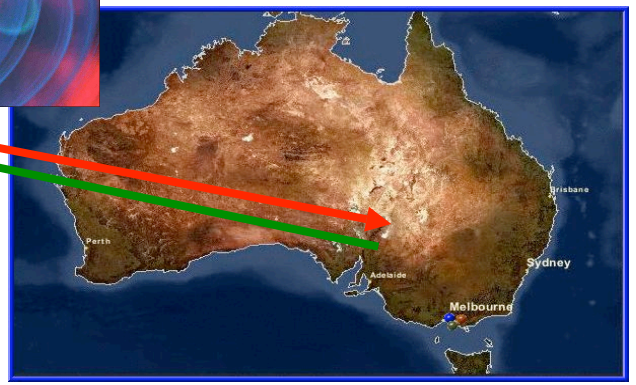
Task Farming, Spawning & Migration

Main Cactus BH Simulation starts in Berkeley





zens of small jobs sent to test parameters



Data returned for main job

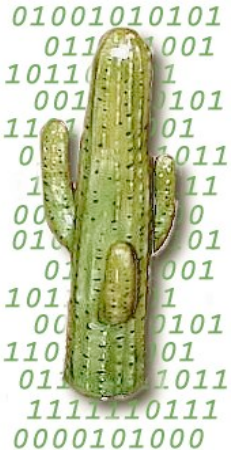
Huge job generates remote data to be visualized in Baltimore

All Require a Common Infrastructure

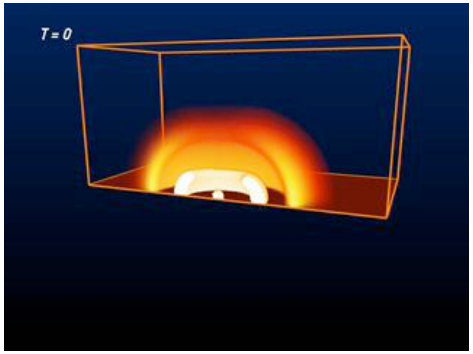
- Common Needs Driven by the Science/Engineering
  - Large Number of Sensors / Instruments
  - Daily Generation of Large Data Sets
  - Data is on Multiple Length and Time Scales
  - Automatic Archiving in Distributed Federated Repositories
  - Large Community of End Users
  - Multi-Megapixel and Immersive Visualization
  - Collaborative Analysis From Multiple Sites
  - *Complex Simulations Needed to Interpret Data*
- Will need Optical Networks
  - Communications → Dedicated Lambdas
  - Data → Large Peer-to-Peer Lambda Attached Storage
  - FAQ: Will usage policies allow scientists to use them?

Source: Smarr

Developing Advanced Applications on Grids



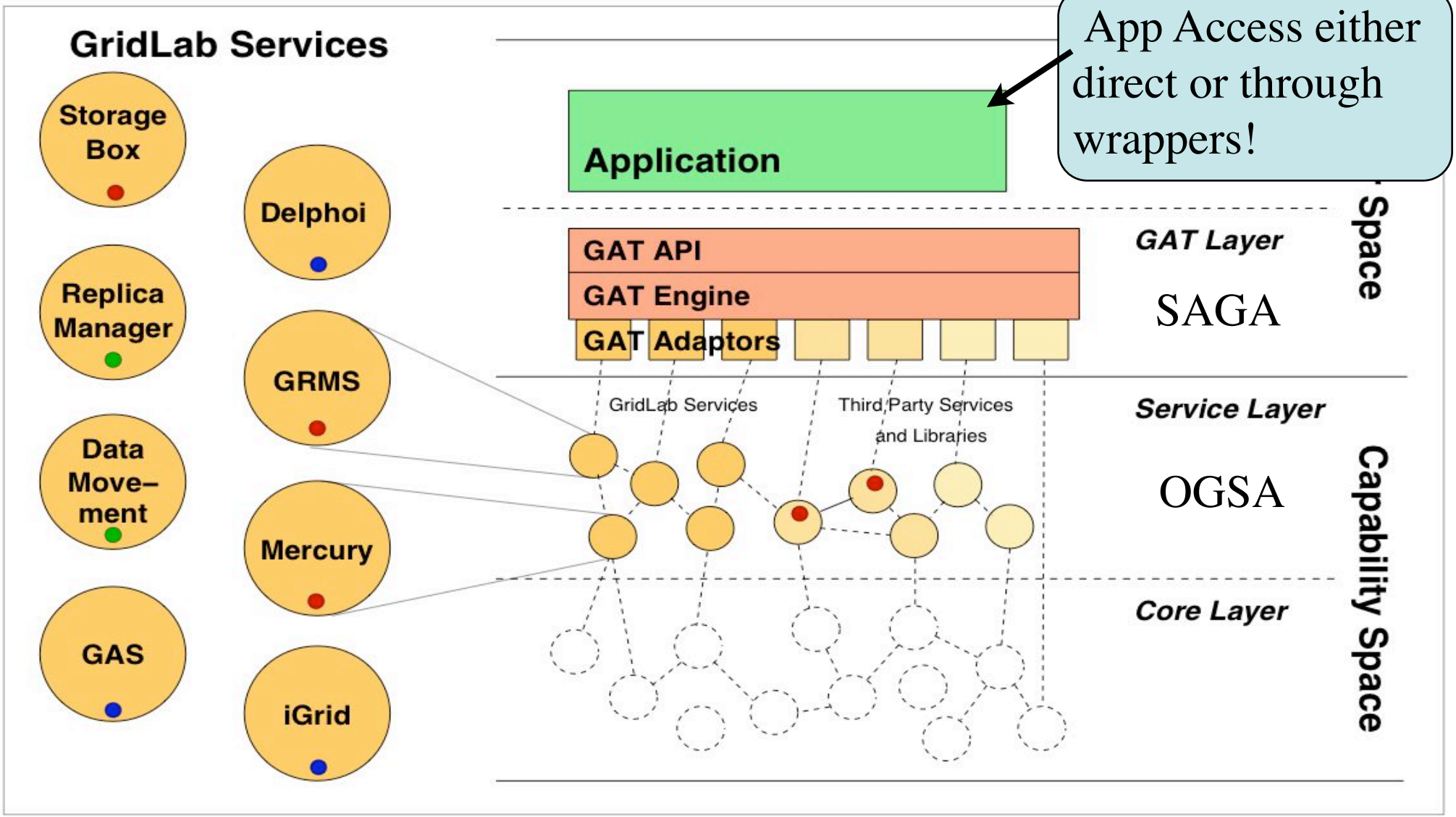
GridLab



GridLab: 5M€ EU Project
New Paradigms

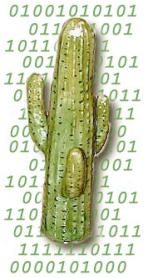
- Code/User/Infrastructure should be aware of environment
  - What *Grid Sevices* are available??
    - Discover resources available NOW, and their current state?
    - What is my allocation?
    - What is the bandwidth/latency between sites?
- Code/User/Infrastructure should be able to make decisions
  - A slow part of my simulation can run asynchronously…spawn it off!
  - New, more powerful resources just became available…migrate there!
  - Machine went down…reconfigure and recover!
  - Need more memory (or less!)…get it by adding (dropping) machines!
- Code/User/Infrastructure should be able to publish to central server for tracking, monitoring, steering…
  - Unexpected event…notify users!
  - Collaborators from around the world all connect, examine simulation.
- Rethink Algorithms: Task farming, Vectors, Pipelines, etc all apply on Grids… The Grid IS your Computer!

GridLab Architecture

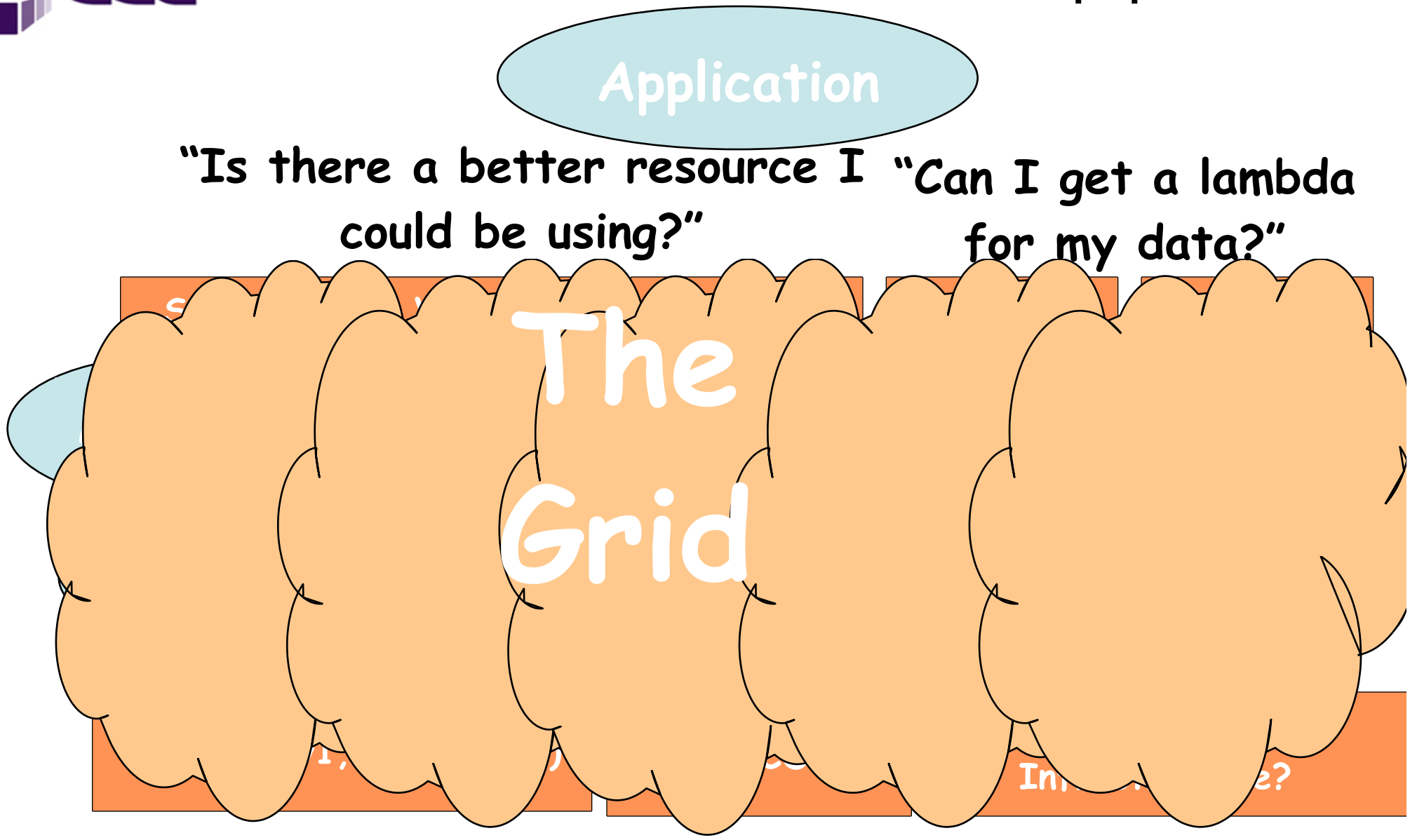


Components Developed to Support New Apps

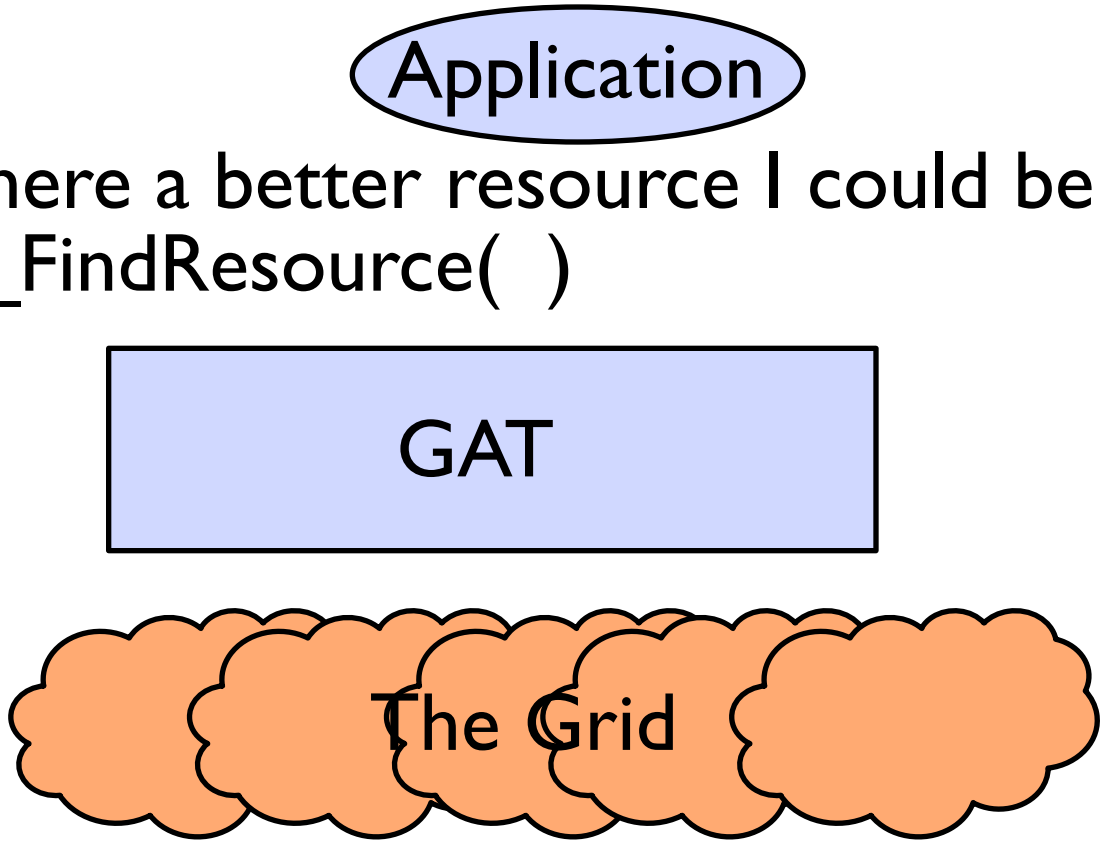
- Core Infrastructure
  - Globus NMI Distribution, Testbeds & Grids (e.g., OSG, TeraGrid)
- Information Servers, Data Replica, Monitoring
- Grid Application Toolkits
  - GAT, SAGA effort in GGF
- Application Frameworks
  - Cactus, Triana, Paramesh, Chombo, Samrai
- Portals
  - Gridsphere Framework, Open GCE, portlets



How to build Grid Apps?

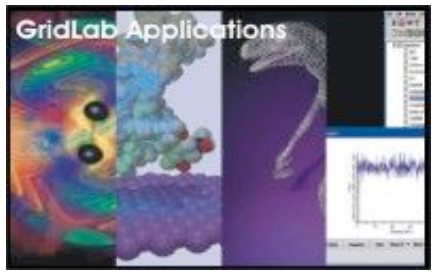


Hiding Complexity with Toolkits

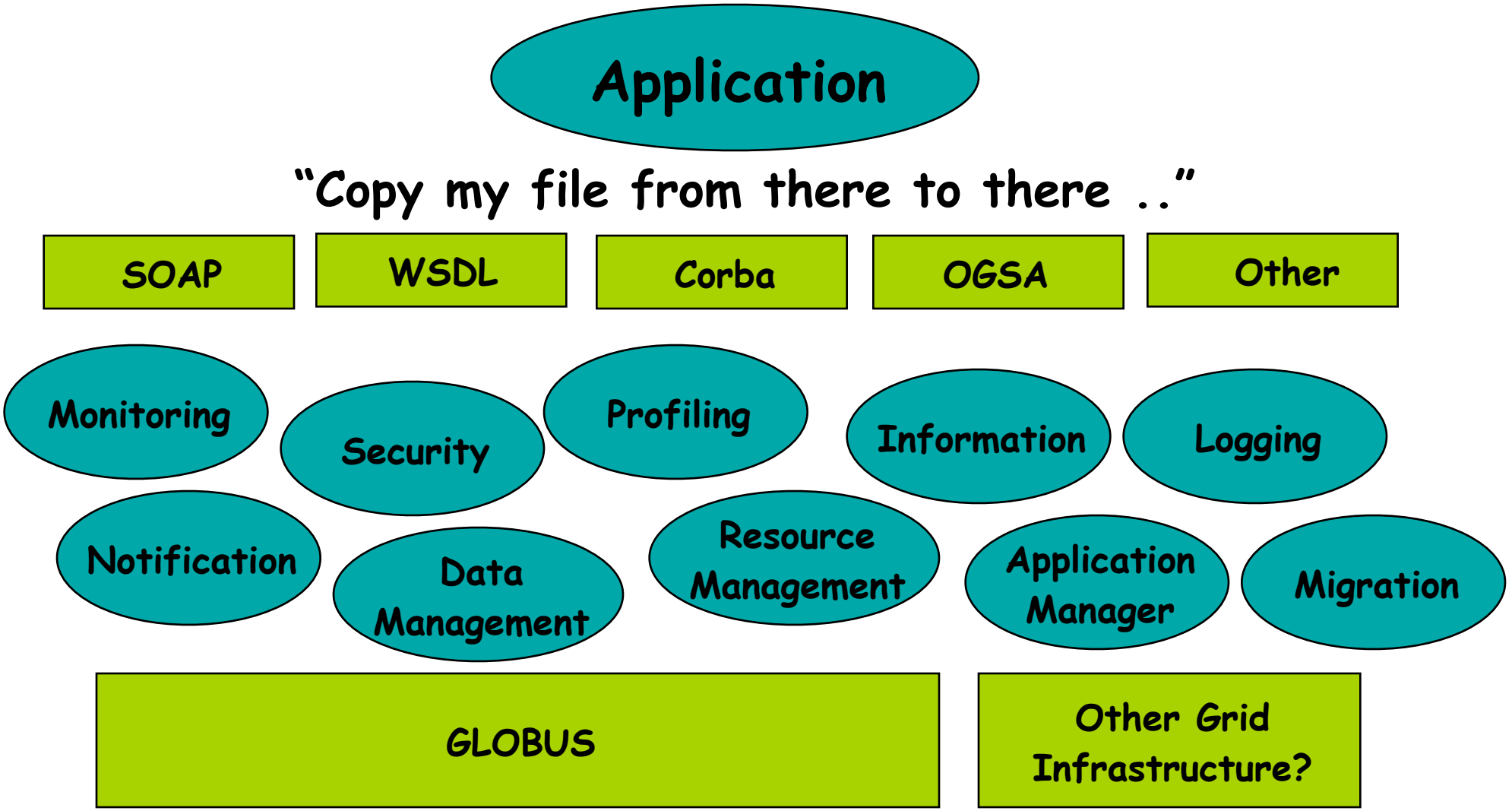


Grid Application Toolkit (GAT) Motivation

- Grids and Grid middleware are everywhere
  - No existing *application oriented standards* for Grid services
  - Applications should be independent of specific infrastructure
- Grid application development lags
  - Missing or immature grid services
  - Changing environment
  - Different and evolving interfaces to the “grid”
  - Interfaces are not aimed at scientific application developers
- Application developers will accept Grid computing paradigm only slowly



Example: Remote File Copying



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++.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;
}
```

GAT Implementation

- Java, C, C++ version fully implemented
- Python, Perl, Fortran to follow
- Focus: high level, portability, lightness, flexibility, adaptivity
- Adaptors
  - Multiple adaptors can be loaded at runtime
  - Local adaptors done (cp, mv, fork, rm, sockets, etc)
  - Many remote service adaptors finished (GRAM, gridftp, scp, GRMS, Mercury, Unicore, etc)
    - Others started (Unicore, Globus, Condor, DRMAA, Sun Grid Engine)
    - Service developers take notice: prepare adaptors!
- Cactus Toolkit: GAT routines available

GGF SAGA-RG

A. Merzky, T. Goodale. S. Newhouse, et al

- GAT evolves into GGF standard
  - Numerous attempts to address: GAT most ambitious, but also CoG, DRMAA, GridRPC, GridCPR, many others
- SAGA: Simple API for Grid Applications
  - Bringing all these efforts together through single API spec
  - Chicago, Berlin, Brussels, LSU, Berkeley, Seoul, and Chicago in June, 2005
- GGF focussing now on standardization
  - SAGA API spec done
  - Much momentum in SAGA now

Finally

- Optical Networks, Grids promise new ways of computing
  - Networks need App toolkits, reasonable cost model
- Standards developing
  - 15 years ago: parallel computing drove interconnects, HPF, MPI
  - Now: 2 levels...OGSA grid services, SAGA for apps
- GridLab: *www.gridlab.org*
- Grid Application Toolkit: *www.gridlab.org/GAT*
  - Documentation, publications, software download
- Cactus Computational Toolkit: *www.cactuscode.org*
- GGF "Simple API for Grid Applications" (SAGA)
  - *forge.gridforum.org/projects/saga-rg*