Dynamic Applications on Grids
Computing ---> Data, Metadata, Networks

Ed Seidel
Center for Computation & Technology
Louisiana State University
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
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
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
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): DDDAS
  - Apps distributed, spawned, task farmed, controlled, monitored with other apps/people
Collaborations for Complex Problems

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

**EU Astrophysics Network**
- 10 EU Institutions
- 3 years
- Continue these problems

**NSF Black Hole Grand Challenge**
- 8 US Institutions
- 5 years
- Attack colliding black hole problem

**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
Recommendations from PITAC

• Universities must significantly change organizational structures: multidisciplinary & collaborative research to remain competitive in global science.

• Federal investments must rebalance to:
  - **Software**: create reliable, easy to use, scalable software that will enable scientists to focus on discovery: Software Crisis
  - **Hardware**: develop, prototype, evaluate new hardware architectures to deliver larger peak and sustained performance at the petaflop level for scientific apps
  - **Data**: focus on data-intensive solutions to address the coming data explosion with advances in sensors and sensor networks

Source: President’s Information Technology Advisory Committee

www.nitrd.gov/pitac
The NCSA-CCT Interdisciplinary Model

Core CS: Gabrielle Allen

Computational Frameworks  Grid, Computing  Comp Math  Scientific Computing  Networks, Sensors, GIS

Coast to Cosmos (C2C): Jorge Pullin

Numerical Relativity  CFD  Coastal/Climate Modeling  Geosciences & Petro engineering  Astro

Visualization, Interaction and Digital Arts (VIDA): Steve Peck

Scientific Viz  Human Computer Interaction  Digital Audio & Music  Collaborative Environments

Material World

Computational Chemistry  Computational Biology

Business

Financial Modeling?  Virtual Organizations?  Scheduling?

Creating Forecast Model for Coastal Erosion using LONI
Huge Black Hole Collision Simulation

CENTER FOR COMPUTATION & TECHNOLOGY AT LOUISIANA STATE UNIVERSITY
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? Data Archives! Metadata!
  – 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! Grids evolve: DDDAS

• Complex infrastructure accessed through common toolkits
  – Everyone should not reinvent the wheel (especially physicists)
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
- Toolkits: Einstein Toolkit, CFD Toolkit, Bioinformatics Toolkit
- International developer/user base
  - We have openings for developers!!
Dynamic Adaptive Distributed Computation
(T.Dramlitsch, with Argonne/U.Chicago)

These experiments:
- Einstein Equations (but could be any Cactus application)
  Achieved:
  - First runs: 15% scaling
  - With new techniques: 70-85% scaling, ~250GF

SDSC IBM SP
1024 procs
5x12x17 =1020

NCSA Origin Array
256+128+128
5x12x(4+2+2) =480

GigE:100MB/sec
(But only 2.5MB/sec)

Dynamic Adaptation: Number of ghostzones, compression, ...

Won “Gordon Bell Prize” (Supercomputing 2001, Denver)
Remote Viz data

Streaming HDF5

Autodownsampling

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

Amira

Interacting with Jobs using Cactus
Notification and Information

Replica Catalog

SMS Server

GridSphere Portal

"The Grid"

Mail Server

www.cactuscode.org

www.gridlab.org
Cactus Portal

- **GridSphere**-based, JSR 168 compliant portal.
- Monitors Cactus applications by sending metadata to the portal at application *startup, shutdown, and regular user-defined intervals*.
- Keeps users updated through *instant messages* and/or *e-mail*.
- Has facilities for *sharing, viewing, editing* and *comparing* parameter files.
- Modular design, allowing new Cactus (or other) applications to easily customize metadata.
Computers as Petabyte Generators
(Black holes and more…)

• Crude vacuum BH collisions today
  – $10^{15}$ Flops, Tbytes output, vastly downsamped!
• 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
• Need metadata to describe simulations
• 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!
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 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
- DDDDAS: injecting data into the above, feed back to experiment
- Model-Model coupling: very important in DDDAS
Already did a lot of this!

Spawning on ARG Testbed

Main Cactus BH Simulation starts here

All analysis tasks spawned automatically to free resources worldwide

User only has to invoke “Spawner” thorn…
Already did a lot of this!

Spawning on ARG Testbed

Main Cactus BH Simulation
starts here

All analysis tasks spawned automatically to free resources worldwide

User only has to invoke “Spawner” thorn…
Already did a lot of this!

Spawning on ARG Testbed

Main Cactus BH Simulation
starts here

All analysis tasks spawned
automatically to free resources
worldwide

User only has to invoke
“Spawner” thorn…
Already did a lot of this! Spawning on ARG Testbed

Main Cactus BH Simulation starts here

All analysis tasks spawned automatically to free resources worldwide

User only has to invoke “Spawner” thorn…
Already did a lot of this!
Spawning on ARG Testbed

Main Cactus BH Simulation
starts here

All analysis tasks spawned
descriptively to free resources
worldwide

User only has to invoke
“Spawner” thorn…
Already did a lot of this!
Spawning on ARG Testbed

Main Cactus BH Simulation
starts here

All analysis tasks spawned
automatically to free resources
worldwide

User only has to invoke
“Spawner” thorn…
Task Farming, Spawning & Migration
Main Cactus BH Simulation starts in Berkeley
Main Cactus BH Simulation starts in Berkeley
Task Farming, Spawning & Migration

Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters
Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters
Task Farming, Spawning & Migration

Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters
Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters
Task Farming, Spawning & Migration

Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters
Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters
Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters
Task Farming, Spawning & Migration

Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters
Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters

Data returned for main job
Task Farming, Spawning & Migration

Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters

Data returned for main job
Task Farming, Spawning & Migration

Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters

Data returned for main job
Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters

Data returned for main job
Task Farming, Spawning & Migration

Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters

Data returned for main job
Main Cactus BH Simulation starts in Berkeley

Dozens of small jobs sent out to test parameters

Data returned for main job

Huge job generates remote data to be visualized in Baltimore
Task Farming, Spawning & Migration

Main Cactus BH Simulation starts in Berkeley

Data returned for main job
Huge job generates remote data to be visualized in Baltimore

Dozens of small jobs sent out to test parameters
Main Cactus BH Simulation starts in Berkeley

Tens of small jobs sent out to test parameters

Data returned for main job

Huge job generates remote data to be visualized in Baltimore
DDDAS Scenarios (Gabrielle Allen)

- **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, DynaCode (Coastal Modeling)**
  - Data coming in from sensors all over Gulf Realtime Operational Grid
  - Feeds into models on Grid sites
  - Algorithms invoked dynamically
Emergency Forecast

Katrina Adv. 25
11 PM EDT SUN AUG 28 2005
Very accurate forecasts are possible; reliability can be improved.
Emergency Forecast

Very accurate forecasts are possible; reliability can be improved.

Used to notify officials, FEMA, State, Governor. Email, web, phone: can do better!

Katrina Adv. 25
11 PM EDT SUN AUG 28 2005
SURA Coastal Ocean Observing Program (SCOOP)

- Integrating data from regional observing systems for realtime coastal forecasts in SE
- Coastal modelers, computer scientists
  - couple models
  - provide data solutions
  - deploy ensembles of models on the Grid
  - assemble realtime results with GIS
DynaCode

• Focus on scenarios:
  – Hurricane ensemble modeling
    • Coupling ocean circulation, storm surge, wave generation models for the Gulf
    • Notifications from NHC trigger customized ensemble hurricane models (surge/wind/wave), sensors verify, guide dynamic ensembles
    • Event driven, dynamic component framework with algorithm selection, optimization tools, workflow, data assimilation, result validation with sensor/satellite.
  – Ecological restoration and control
    • Breton Sound diversion, control structure to allow Mississippi to flow into wetlands
    • Coupled models (hydrodynamic, salinity, geomorphic, sediment) control diversion, sensors/wind fields inject real time data.
Dynamic, Event Driven Computing

- Archive
- Ensemble
- Analytical
- Winds
- Application Manager
- Data Assimilator
- Sensor Arrays
Dynamic, Event Driven Computing

Archive

Ensemble
Analytical
Winds

Application Manager

Data Assimilator

Sensor Arrays
Dynamic, Event Driven Computing

Hurricane Notification

Use Early warning Start Preparation For Emergency Runs.

Early Warning Track Info.

Ensemble Analytical Winds

Archive

Application Manager

Based on Early Warning Vacate Supercomputer

Data Assimilator

Sensor Arrays
Dynamic, Event Driven Computing

Hurricane Notification

Archive

Ensemble Analytical Winds

Retrieve Data files as they become available

Application Manager

Data Assimilator

Sensor Arrays
Dynamic, Event Driven Computing

Hurricane Notification

Ensemble Analytical Winds

Archive

Retrieve Data files as they become available

Application Manager

Data Assimilator

Dynamic Model Coupling!
Dynamic, Event Driven Computing

Hurricane Notification

Ensemble Analytical Winds

Archive

Retrieve Data files as they become available

Application Manager

Data Assimilator

Sensor Arrays
Dynamic, Event Driven Computing

- Hurricane Notification
- Archive
- Ensemble
- Analytical
- Winds
- Application Manager
- Data Assimilator
- Sensor Arrays
Dynamic, Event Driven Computing

Notification!
Emergency Response!

Application Manager

Archive

Data Assimilator

Results Dissemination

Hurricane Notification

Sensor Arrays
Dynamic, Event Driven Computing

Data Driven!

DDDAS

Application Manager

Archive

Ensemble
Analytical

Results Dissemination

Sensor Arrays

Data Assimilator
Dynamic, Event Driven Computing

Hurricane Notification

Archive

Ensemble
Analytical
Winds

Application Manager

Data Assimilator

Results Dissemination

Results Verification

Sensor Arrays
Coordinated use of compute and network resources

Define the network through the set of services that it can offer

MCNC (lead), LSU, RENCI, NCSU, Cisco, AT&T, Calient Networks

10Gig capacity testbed running over LONI and NLR
**Scenario**
- TBs data, must be interactively viz’d/analyzed by international collaboration

**Distributed Data and Viz Services**
- Data larger than memory, disk slow: streaming data servers
- Co-scheduling machines, network for collaborative event
- Lambda Provisioning

**High Resolution: not science w/o details, analysis**
- Uncompressed HD stream: near latency-free remote viz at 1.5Gbit
- Video for colleagues
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?
Developing Advanced Applications on Grids
GridLab Architecture

GridLab Services
- Storage Box
- Replica Manager
- Data Movement
- Mercury
- GAS
- iGrid

Application Layer
- Application
  - GAT API
  - GAT Engine
  - GAT Adaptors

User Space
- SAGA

Service Layer
- OGSA

Capability Space
- GridLab Services
- Third Party Services and Libraries
GridLab Architecture

App Access either direct or through wrappers!
How to build Grid Apps?

"Is there a better resource I could be using?"

"Can I get a lambda for my data?"
How to build Grid Apps?

"Is there a better resource I could be using?"  "Can I get a lambda for my data?"

SOAP  WSDL  CORBA  OGSA  Other

Monitoring  Security  Profiling  Information  Logging

Notification  Data Management  Resource Management  Application Management  Migration

GLOBUS (v1, 2, 3, 4, ...)

UNICORE

Other Grid Infrastructure?
Hiding Complexity with Toolkits

Application

“Is there a better resource I could be using?”
GAT_FindResource( )

GAT

The Grid
Example: Remote File Copying

"Copy my file from there to there .."
Copy a File: GASS

```c
int RemoteFile::GetFile (char const* source, char const* target) {
    globus_url_t source_url;
    globus_ftp_client_operationattr_t source_ftp_attr;
    globus_gass_transfer_requestattr_t source_gass_attr;
    globus_gass_copy_attr_t source_gass_copy_attr;
    globus_gass_copy_handleattr_t gass_copy_handleattr;
    globus_io_attr_t io_attr;
    output_file = -1;

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

    if (source_url.scheme_type != GLOBUS_URL_SCHEME_GSIFTP ||
        source_url.scheme_type != GLOBUS_URL_SCHEME_FTP) {
        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 "quote)s
"quote)n", target);
            return (-1);
        }

        globus_gass_copy_handleattr_init  (&gass_copy_handleattr);
        globus_gass_copy_attr_init        (&source_gass_copy_attr);
        globus_ftp_client_attr_init      (&ftp_client_attr);
        globus_io_fileattr_init          (&io_attr);
        globus_gass_copy_attribute_init  (&io_attr);
        globus_gass_copy_attr_set_io     (&source_gass_copy_attr, &io_attr);
        globus_gass_copy_handleattr_set_ftp_attr

        result = globus_gass_copy_register_url_to_handle (     
                &gass_copy_handle, (char*)source_URL,     
                &source_gass_copy_attr, &io_attr, 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);
    }
```
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));
            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()));
            int requestid = rftPort.start();
            System.out.println("Request id: " + requestid);
            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);
            }
        } catch (java.io.FileNotFoundException fnfe) {
        } finally {
            reader.close();
        }
    } catch (Exception e) {
        System.err.println("Exception: "+e);
    }
}

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);
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(TimeSpan.ChronoUnit.valueOf((String)requestData.elementAt(5)).booleanValue());
int requestid = rftPort.start();
System.out.println("Request id: " + requestid);
} catch (Exception e) {
    System.err.println(MessageUtils.toString(e));
}
#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;
}
GGF SAGA-WG
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, Chicago, Boston

• GGF focusing now on standardization
  – SAGA API spec done
  – Much momentum in SAGA now
Finally

- Optical Networks, Grids promise new ways of computing
  - More than computing: Network, Data services critical
- Complex applications require comprehensive services
  - Event Driven: many examples. Hurricanes, earthquakes, tornados, explosions, ...
  - Much of this can work now as prototypes
- Standards developing
  - 15 years ago: parallel computing drove interconnects, HPF, MPI
  - Now: 2 levels...OGSA grid services, SAGA for apps
- Cactus Computational Toolkit: www.cactuscode.org
- GGF “Simple API for Grid Applications” (SAGA)