

# Building Cyberinfrastructure-Enabled and Community-Centric Science Gateway Applications

**Yan Liu<sup>1</sup>, Shaowen Wang<sup>1</sup>, Raminder Singh<sup>2</sup>,  
Suresh Marru<sup>2</sup>, Marlon Pierce<sup>2</sup>, Nancy Wilkins-Diehr<sup>3</sup>**

**<sup>1</sup>CyberInfrastructure and  
Geospatial Information  
Laboratory  
and  
National Center for  
Supercomputing Applications  
University of Illinois at  
Urbana-Champaign**

**<sup>2</sup>Pervasive Technology  
Institute  
Indiana University**

**<sup>3</sup>San Diego Supercomputer  
Center  
University of California, San  
Diego**

***TeraGrid'11, July 18, 2011  
Salt Lake City, Utah***



TeraGrid™

# Objectives

- Use TeraGrid to support domain-specific scientific computing
- Develop Grid-enabled applications to access TeraGrid capabilities
- Create gateway application Web service
- Build user-centric Web 2.0 environment as science gateway user environment

<http://simplegrid.org/>

<http://gisolve.org>

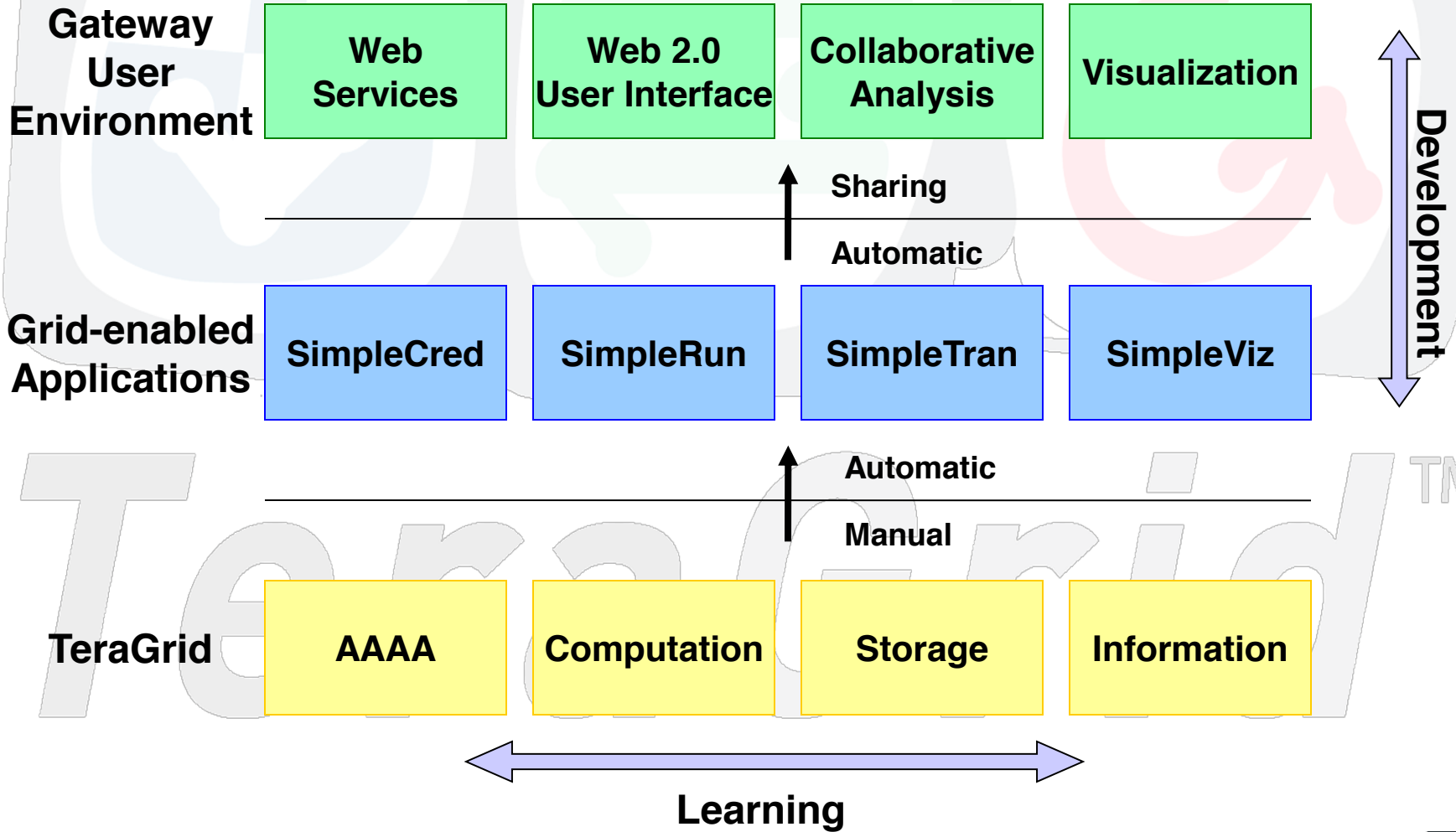
<http://collab-ogce.org>

TeraGrid™



TeraGrid™

# SimpleGrid Overview



TeraGrid™

# Learning Curve

- **Case study: convert a domain-specific application into TeraGrid-enabled science gateway application**
  - Spatial interpolation code in GIScience domain
  - Simplified from a production GISolve application
- **Three-stage learning**
  - Leveraging TeraGrid resources for computation
  - Streamlining Grid-based scientific computing as Grid-enabled Web services
  - Building a simple & user-friendly science gateway interface using Web 2.0 technologies



TeraGrid™

# Progressive Learning

- **Access TeraGrid resources**

- Accounts and allocations
- Computing
- Data storage
- Information services

- **Develop Grid-enabled applications**

- Globus Toolkit programming for TeraGrid access
- Encapsulating Grid-enabled applications as Web services

- **Build a simple science gateway user environment**

- Web 2.0 interface for user interactions and community application integration
- Computation management in science gateway using OGCE



TeraGrid™

# Development Curve

- TeraGrid resources and services for domain-specific computation
  - Computing on TeraGrid directly using tools and middleware
- Grid-enabled application
  - Globus Toolkit programming for TeraGrid access
  - Application-specific SimpleGrid API
  - Axis2-based Web service development
- Reusable components for development
  - SimpleGrid APIs
    - Designed to be simple
  - Gateway computation management modules
  - Useful links for further study
- Science gateway
  - Web 2.0 user environment development
    - JavaScript programming on user interface components (Yahoo UI)
    - Ajax communications
    - Php-based server code programming
  - A straightforward GIScience (Geographic Information Science) collaborative application workflow



TeraGrid™

# Application Examples

- **Applications**

- Test application
- Real-world application: DMS spatial interpolation
  - Widely used in community research and education

- **Application package**

- Binary executables
- Sample datasets

- **Build user-friendly science gateway interfaces for scientists to perform analysis transparently on TeraGrid**

- Developing user-friendly gateway Web interface to meet user interaction requirements
- Using TeraGrid tools and services for analysis computation on TeraGrid resources
- Streamlining the computation process by developing the Grid-enabled application Web services



TeraGrid™

# Outline

- Part I: SimpleGrid gateway and gadgets demonstration
- Part II: Command line: TeraGrid resources and services
- Part III: Application: Grid-enabled analysis using SimpleGrid API
- Part IV: Services: application and computation
- Part V: Sharing: SimpleGrid user environment development



TeraGrid™



# Part I: SimpleGrid Gateway and Gadgets Demonstration

- **Science gateways to TeraGrid**

- Provide an online access points to TeraGrid
- Aggregate domain science application-level capabilities
- Hide the complexity of using TeraGrid
- Web-based user environment is commonly used to meet such needs

- **Focus**

- Simplified user access through gateway user environment
- Provide technological demonstrations for gateway development
- Provide transparent access to TeraGrid for domain scientists
- Build application-oriented workflow for easy access by users

<http://www.teragrid.org/gateways/>

 **NCSA**



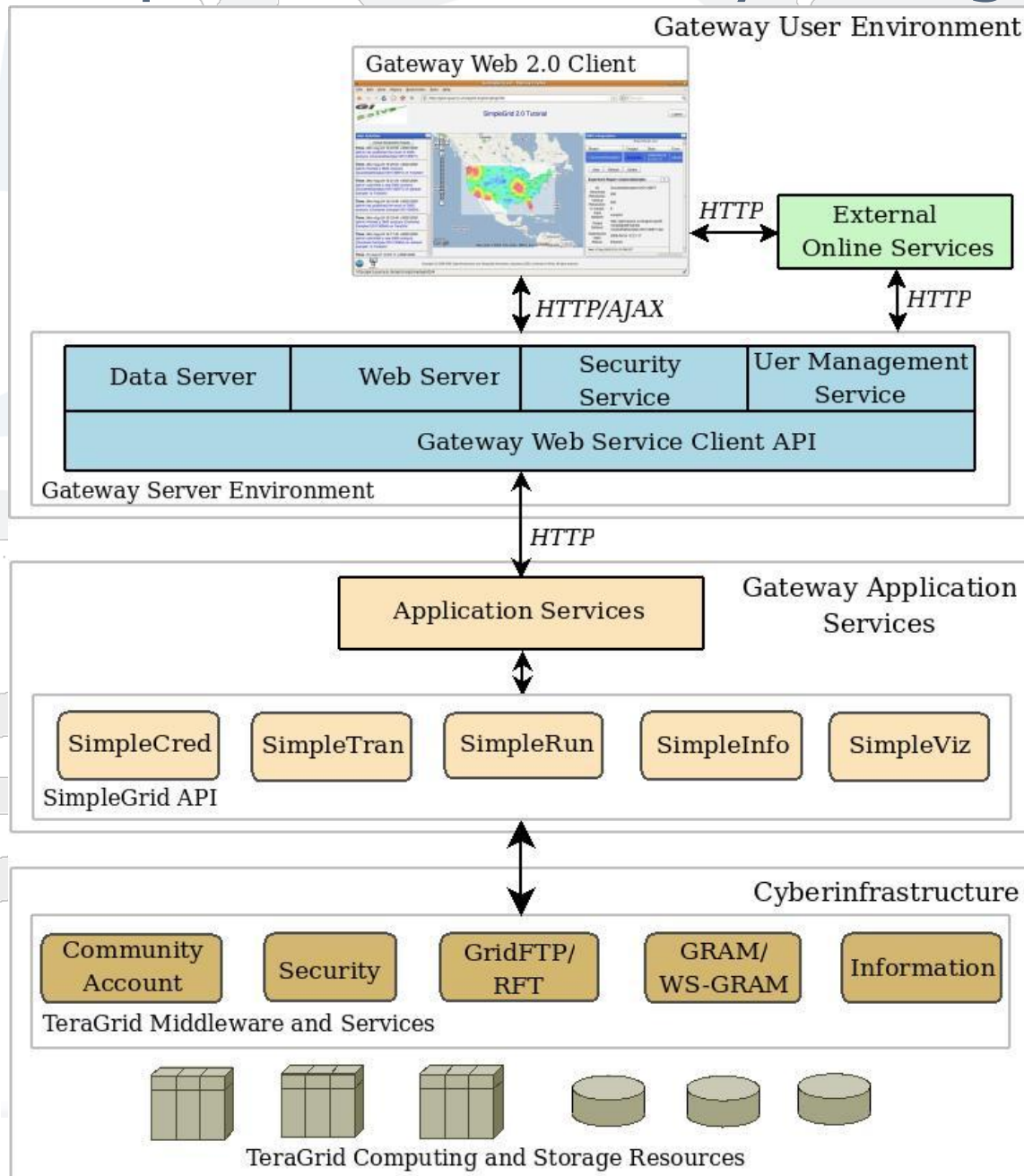
INDIANA UNIVERSITY

**SDSC**



TeraGrid™

# SimpleGrid Gateway Design



SC



TeraGrid™

# DMS Analysis Using Gateway User Environment

- **Client:** click “New” to create a DMS job
- **Server:** job submission
  - Invoking DMS Web service for job submission
  - Recording job in database
- **Client:** automatic job status refresh through YUI timer
- **Server:** handling job status refresh request
  - Invoking DMS Web service for job status
  - Updating database
  - Posting Twitter updates
- **Client:** receiving Twitter updates periodically
- **Client:** visualization
  - Drag&drop
  - Client-side JavaScript call for Google Map overlay creation and display

TeraGrid™



TeraGrid™

# DMS Analysis in OGCE Gadget Container

- OGCE introduction
- SimpleGrid gadgets
  - Job gadget
  - Visualization gadget
- Inter-gadget communication
- Authentication among gadgets, container, and gadget services

# TeraGrid™



TeraGrid™



*TeraGrid*<sup>TM</sup>



TeraGrid<sup>TM</sup>

# Exercise

- Science gateway account
- Conduct a DMS analysis
  - Name the analysis as “dms-yourname”
  - Submit an analysis
  - Follow SimpleGrid tweets
  - Refresh job status
  - Drag the finished job to Google map

*TeraGrid*™



TeraGrid™

# Demystifying the SimpleGrid Gateway Development

## • User environment development

- Client side-centric user interactions
  - Contact server if necessary
- Web 2.0 rich client that leverages JavaScript/Ajax libraries for Web content rendering, communications, interactions with external online services (e.g., Google Map and Twitter)
- Web server acts as an information provider and communication proxy

## • TeraGrid functions

- TeraGrid access is hidden from user environment
- Integrated into backend service-oriented computing architecture

## • Grid-enable analysis application service development

- Simplified programming interface
- Streamlined service creation
- Gadget development using OGCE gadget container



TeraGrid™

# Tutorial Environment

- Tutorial server
  - gw3.quarry.iu.teragrid.org
- Tutorial server software
  - SSH client and server
  - Globus Toolkit 4.0.1 client tools
  - Java J2SE 1.6
  - Ant 1.6.5
  - Apache httpd 2.2
  - Tomcat 5.5.27
  - Axis2 1.5.1
  - YUI 2.8.1
  - PHP 5.1.6
  - MySQL server 5.0.45
- TeraGrid sites
  - TACC Lonestar

TeraGrid™



TeraGrid™



# Tutorial Accounts

- SimpleGrid community account
  - simpleg
- Tutorial server Web and Unix accounts
  - train1 – train9
- Password: tg11GWAPP

*TeraGrid*<sup>TM</sup>



TeraGrid<sup>™</sup>

## Part II: Command line: TeraGrid resources and services

- **Questions to answer**

- Where are your home directories on all 3 TeraGrid sites?
- What is your grid certificate identity?
- How do you create a grid proxy with a specified duration using MyProxy?
- Are you able to transfer a sample dataset to a specified TeraGrid site?
- How to submit a job and return immediately without waiting for its completion?

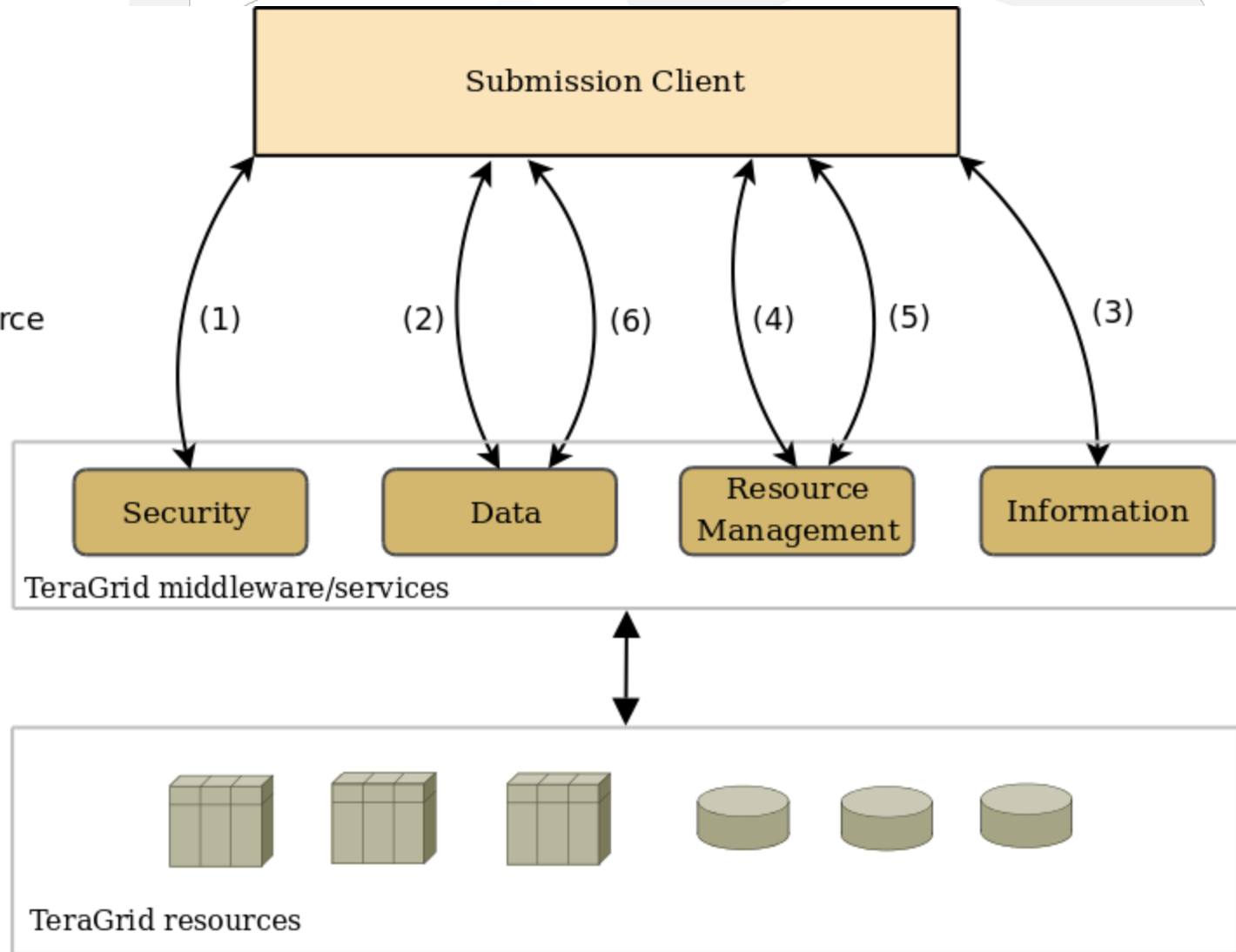
TeraGrid™



TeraGrid™

# Computation Flow

- (1) Get Grid credential
- (2) Transfer datasets
- (3) Select TeraGrid resource
- (4) Submit Grid job
- (5) Monitor job status
- (6) Transfer results



TeraGrid™

# Technologies

## • TeraGrid services

- Allocation and accounts
  - TeraGrid User Portal (TGUP)
- Software environment
  - CTSS (Coordinated TeraGrid Software and Services)
- Resource management
  - Using GRAM/WS-GRAM to access TeraGrid HPC clusters
- Data storage
  - Using GridFTP/RFT to access TeraGrid storage services
- Information
  - Resource selection through TeraGrid information services
- Security
  - Grid security
  - MyProxy

## – Technical support

- Helpdesk: [help@teragrid.org](mailto:help@teragrid.org)



TeraGrid™

# TeraGrid Allocation and Accounts

DEMO: TeraGrid User Portal

# TeraGrid™



TeraGrid™

# Computing Environment on TeraGrid

- Login to a TeraGrid cluster's head node
  - SSH
  - TeraGrid user portal -> My TeraGrid -> SSH Terminal

```
$ ssh trainXX@honest1.ncsa.uiuc.edu
```

- Check your software environment by looking at `$HOME/.soft`

```
## content of your .soft setting  
@teragrid-basic  
# TeraGrid wide Globus 4 and Grid software suite  
@globus-4.0  
# Platform recommended development software suite  
@teragrid-dev
```

- Check resource allocation status

```
$ tgusage --account=TG-***
```



TeraGrid™

# Exercise: Setup Development Environment

- Login to tutorial server
  - SSH login: `username@${tutorial_server}`
- Environment setup

```
cd ~  
# setup grid environment  
./opt/software/tg-client-env.sh
```

*TeraGrid*<sup>TM</sup>



TeraGrid™

# Get a Grid Proxy on Client

```
$ myproxy-logon -l gisolve -t 100 -s myproxy.teragrid.org -p 7514
```

```
Enter MyProxy pass phrase:
```

```
A credential has been received for user gisolve in /tmp/x509up_u502.
```

```
$ grid-proxy-info
```

```
subject : /C=US/O=National Center for Supercomputing Applications/CN=Gisolve  
Community User
```

```
issuer  : /C=US/O=National Center for Supercomputing  
Applications/CN=Certification Authority
```

```
identity : /C=US/O=National Center for Supercomputing Applications/CN=Gisolve  
Community User
```

```
type    : end entity credential
```

```
strength : 1024 bits
```

```
path    : /tmp/x509up_u502
```

```
timeleft : 99:59:54 (4.1 days)
```

<http://teragrid.org/userinfo/access/>

<http://myproxy.ncsa.uiuc.edu>



TeraGrid™



# Transfer a Sample Dataset

```
$ globus-url-copy file://$HOME/simplegrid2/tmp/sample  
${gridftp_contact}/~/sample1
```

```
$ ssh trainXX@honest1.ncsa.uiuc.edu
```

```
tg-login2 ac/trainXX> ls -l ./tutorial/sample1
```

```
-rw-r--r--  1 trainXX  ipt      1116491 2009-05-29 21:28 sample1
```

- TeraGrid storage resources

- Storage associated with cluster
- Storage system
  - GPFS-WAN
    - Access: Grid-FTP & UberFTP

[http://teragrid.org/userinfo/data/transfer\\_location.php](http://teragrid.org/userinfo/data/transfer_location.php)

<http://teragrid.org/userinfo/data/gridftp.php>

<https://www.teragrid.org/web/user-support/storage>



TeraGrid™

# GRAM Job Submission

- Compose job submission file in RSL format

```
&(jobType="single")
(count=1)
(host_count="1")
(project=TG-SES090019)
(directory="/home/gisolve/gisolve/DMS/release")
(executable=/home/gisolve/gisolve/DMS/release/process.pl)
(arguments="500" "500" "20" "/home/gisolve/sample1" "/home/gisolve/result1")
(stdout="stdout.gisolve.test")
(stderr="stderr.gisolve.test")
```

- Commands

```
$ globusrun -b -f ./gt2.rsl -r grid-hg.ncsa.teragrid.org:2119/jobmanager-pbs
globus_gram_client_callback_allow successful
GRAM Job submission successful
https://tg-grid1.uc.teragrid.org:50004/19686/1180493030/
$ globus-job-status https://tg-grid1.uc.teragrid.org:50004/19686/1180493030/
DONE
```



TeraGrid™

# WS-GRAM Job Submission

- Compose job submission file in RSL XML format

```
<job>
  <factoryEndpoint
    xmlns:gram="http://www.globus.org/namespaces/2004/10/gram/job"
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing">
    <wsa:Address>
      https://grid-hg.ncsa.teragrid.org:8443/wsrf/services/ManagedJobFactoryService
    </wsa:Address>
    <wsa:ReferenceProperties>
      <gram:ResourceID>PBS</gram:ResourceID>
    </wsa:ReferenceProperties>
    </factoryEndpoint>
    <executable>/home/gisolve/gisolve/DMS/release/process.pl</executable>
    <directory>/home/gisolve/gisolve/DMS/release</directory>
    <argument>500</argument>
    <argument>500</argument>
    <argument>20</argument>
    <argument>/home/gisolve/sample1</argument>
    <argument>/home/gisolve/result1</argument>
    <stdout>/users/gisolve/gisolve/DMS/results/stdout.sample1</stdout>
    <stderr>/users/gisolve/gisolve/DMS/results/stderr.sample1</stderr>
  </job>
```

```
$ globusrun-ws -b -submit -f my_rsl_xml
```

<http://teragrid.org/userinfo/jobs/globus.php>



TeraGrid™

# TeraGrid Information Services

- How do I know which TeraGrid site to use for my job submission and data transfers?
  - Check out TeraGrid web site for resource information
  - Use TeraGrid information services

## **\$ ./tginfo gram2 at ncsa**

HTTP Request to: <http://info.teragrid.org/web-apps/csv/tginfo-v1/cli/gram2/at/ncsa>

Name	ResourceID	EndPoint
prews-gram-fork	dtf.ncsa.teragrid.org	grid-hg.ncsa.teragrid.org:2119/jobmanager-fork
prews-gram-pbs	abe.ncsa.teragrid.org	https://grid-abe.ncsa.teragrid.org:2119/jobmanager-pbs
prews-gram-pbs	lincoln.ncsa.teragrid.org	https://grid-abe.ncsa.teragrid.org:2119/jobmanager-pbs
prews-gram-fork	lincoln.ncsa.teragrid.org	https://grid-abe.ncsa.teragrid.org:2119/jobmanager-fork
prews-gram-pbs	cobalt.ncsa.teragrid.org	grid-co.ncsa.teragrid.org:2119/jobmanager-pbs
prews-gram-fork	cobalt.ncsa.teragrid.org	grid-co.ncsa.teragrid.org:2119/jobmanager-fork
prews-gram-fork	abe.ncsa.teragrid.org	https://grid-abe.ncsa.teragrid.org:2119/jobmanager-fork
prews-gram-pbs	dtf.ncsa.teragrid.org	grid-hg.ncsa.teragrid.org:2119/jobmanager-pbs

<http://teragrid.org/userinfo/hardware/resources.php>

<http://info.teragrid.org/>



TeraGrid™

# MPI Job Management in Gateways

- MPI job submission on TeraGrid clusters
  - Abe: qsub examples for PBS (Portable Batch Scheduler)
- How to launch MPI jobs from gateway?
  - Globus RSL
    - jobType: MPI
    - Issue: pre- and post-processing is difficult and error-prone
  - Direct ssh
    - gsissh `{tg-cluster-name} "qsub {my-mpi-qsub-script}"`
    - Issue: no API for gateway development

TeraGrid™



TeraGrid™

# MPI Job Management Discussion

- **Globus GRAM-based solution 1**

- Launch a fork job to do pre-processing and submit MPI job
- Monitor MPI job status through fork jobs
- Transfer results back using GridFTP

- **Globus GRAM-based solution 2**

- Use `jobType=single` with `count=np`
- Issues

- If pre- or post-processing takes long time, other CPUs idle. Not recommended by TeraGrid
- RSL `jobType` interpretation is different on local schedulers
  - [http://bugzilla.globus.org/globus/show\\_bug.cgi?id=3384](http://bugzilla.globus.org/globus/show_bug.cgi?id=3384)



# TeraGrid-Based DMS Analysis Summary

- **Command line practice**

- Globus Toolkit 4.0, MyProxy

- **Typical computing process**

- Request a personal or community account on TeraGrid
- Install DMS executables on three TeraGrid sites
- Prepare a dataset on a local machine
- Transfer a specified dataset to a TeraGrid site (e.g., NCSA)
- Submit a Grid job to the specified TeraGrid site with a parameter value
- The submitted job is scheduled to be executed on one compute node on the specified TeraGrid cluster
- When the job is finished, the analysis result is written into the data directory of DMS installation on the TeraGrid cluster
- Transfer the result back to the local machine
- Visualize the result using the DMS visualization tool

- **Learning experience**

- TeraGrid resource access using Globus Toolkit
- Conduct a scientific computing process using command line tools
- Good for experts, but error-prone and not scalable for science gateway users



TeraGrid™

# Part III: Application: Grid-enabled analysis using SimpleGrid API

- How to do Part II in a programming way?
- Purpose
  - Streamline the access to TeraGrid resources as a Grid-enabled application
  - Build Application Web services for interactions with gateway user environment
- Package
  - Location: \$HOME/simplegrid2/java
  - SimpleGrid API for DMS analysis
  - Libraries
    - All jars from `#{GLOBUS_LOCATION}/lib`
      - Particularly cog-jglobus.jar
    - Apache Axis2

<http://www.globus.org/cog/manual-user.pdf>



TeraGrid™



# Programming Techniques

- Java CoG kit API
- Grid security programming
- MyProxy programming
- SAML attribute-based gateway user authentication programming
- GRAM programming
- GridFTP programming

*TeraGrid*™



TeraGrid™

# SimpleGrid API

- **SimpleCred: Grid credential management**
  - Proxy creation: local and MyProxy
  - Automatic proxy renewal
  - Attribute-based X.509 credential for authorization
- **SimpleTran: Data transfer to/from grids**
  - GridFTP
- **SimpleRun: Grid job management**
  - GRAM and WS-GRAM
  - Batch job submission
- **SimpleViz: Visualization component**
  - Java 2D + Google map (shown later)
- **SimpleInfo: Grid information provider \***
- **Gateway computation management service**

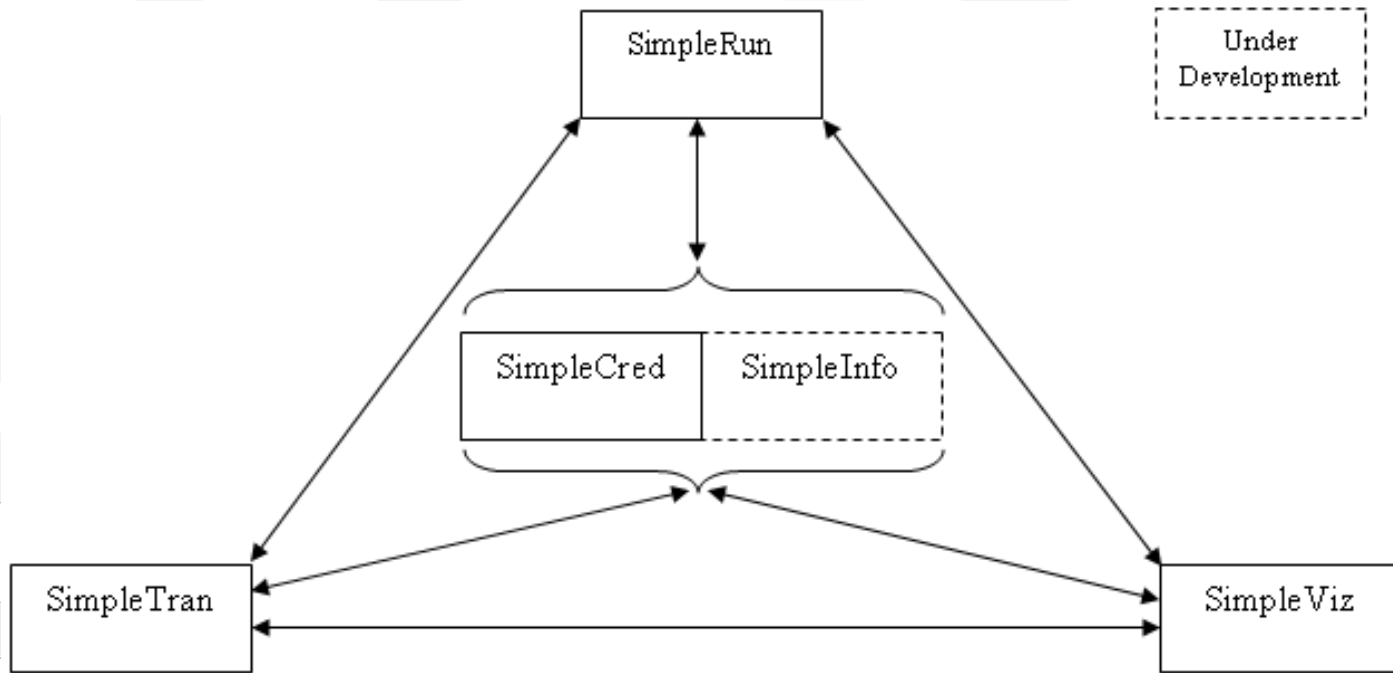


INDIANA UNIVERSITY



TeraGrid™

# SimpleGrid API architecture



- Learning experience

- Java-based Grid-enabled scientific application development
- Application deployment is a constraint for large-scale user community



TeraGrid™

# Get a Grid Proxy

- **Class**

- org.simplegrid.grid.security.SimpleCred

- **Two methods to get a proxy**

- Load an existing valid proxy file

- SimpleCred::load()

- Create a proxy by contacting with a MyProxy server

- SimpleCred::logon()

- **Generalization: SimpleGrid::get()**

- **Export proxy to a file**

- SimpleCred::export()



TeraGrid™

# Transfer a Sample Dataset

- **Class**

- org.simplegrid.grid.data.SimpleTransfer

- **Local <-> remote transfer**

- SimpleTransfer::remote2local()

- SimpleTransfer::local2remote()

- SimpleTransfer::url2remote()

- **Third-party transfer: SimpleTransfer::transfer()**

- **Jglobus programming manual**

- User version 1.1

<http://wiki.cogkit.org>



TeraGrid™

# GRAM Job Submission

- **Class**

- org.simplegrid.grid.job.SimpleRunGT2

- org.simplegrid.grid.job.SimpleRSL

- **RSL generation**

- Application specific

- **GT2 job submission**

- Use batch mode in which a call returns immediately with a job handle

TeraGrid™



TeraGrid™

# GRAM Job Submission: MPI

- **MPI on TeraGrid**

- User “softenv | grep mpi” to see different MPI settings

- **MPI job submission through Globus**

- Specify “count” element in RSL

- “count” is the number of processes to spawn

- Specify “host\_count” element in RSL

- “host\_count” is the number of CPUs for MPI execution

- “count = host\_count” means each CPU runs one MPI process

- Specify “jobType=mpi” in RSL

*TeraGrid*<sup>TM</sup>



TeraGrid™

# WS-GRAM Job Submission

- **Deprecated!!!**
- **Class**
  - org.simplegrid.grid.job.SimpleRunGT4
  - org.simplegrid.grid.job.SimpleRSL
- **RSL XML generation**
  - Application specific
- **GT4 job submission**
  - Use batch mode in which a call returns immediately with a job handle
  - Note that GT4 uses a different globus package `org.globus.exec.*`

<http://www.globus.org/toolkit/docs/4.0/execution/wsgram/developer-index.html#id2563059>



TeraGrid™



# How to Write RSL?

- **RSL schema**

- [http://globus.org/toolkit/docs/4.0/execution/wsgram/schemas/gram\\_job\\_description.html](http://globus.org/toolkit/docs/4.0/execution/wsgram/schemas/gram_job_description.html)

- **GT2 and GT4 schema comparison**

- [http://www.globus.org/toolkit/docs/4.0/execution/wsgram/WS\\_GRAM\\_Migrating\\_Guide.html](http://www.globus.org/toolkit/docs/4.0/execution/wsgram/WS_GRAM_Migrating_Guide.html)

*TeraGrid*<sup>TM</sup>



TeraGrid™

## Exercise 2

- **Test SimpleGrid API**
  - Source simplegrid-env.sh
  - Compile: \$HOME/simplegrid2/bin/AppCompile.sh
  - Run: \$HOME/simplegrid2/bin/AppRun.sh

*TeraGrid*<sup>TM</sup>



TeraGrid™

# Part IV: On-demand service: Creating analysis Web service

## • Why Web services?

- Separating interface and implementation
- Loosely-coupled software components
- Sharing
- Providing on-demand analysis support

## • Web services in Grid computing

- Grid services are Web services
  - Globus services are implemented as Axis2 web services with Grid extension
- TeraGrid employs Grid services at resource provider side
  - Grid services provide a standard way to access TeraGrid for security, resource management, data storage, and information

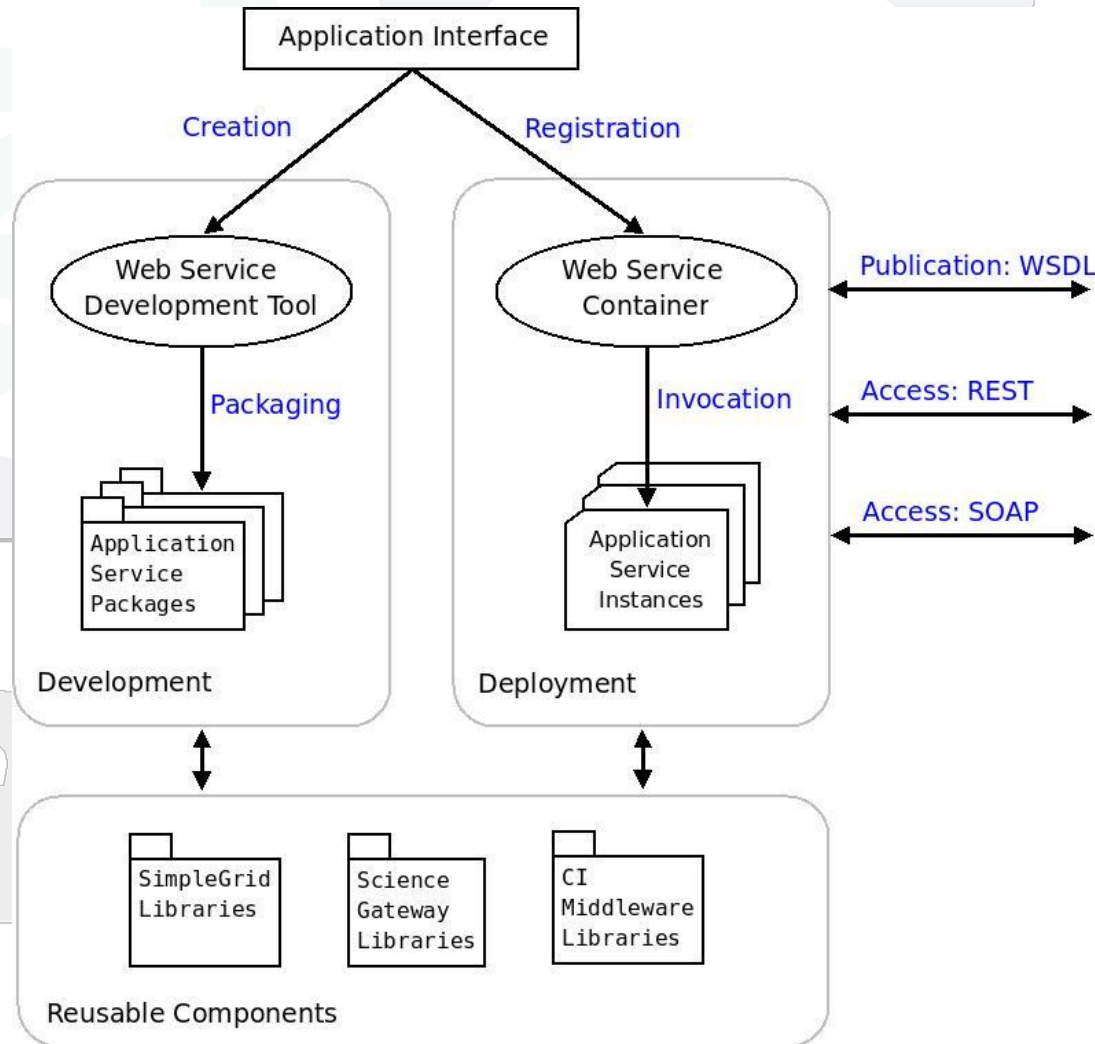


TeraGrid™

# SimpleGrid Web Service Tool

- **ws-gen tool**

- Based on Apache Axis2
- Automatic Web service code generation
- Packaging Web service code and Globus libraries as standalone package



# ws-gen: How It Works

- Generate DMS WSDL (Web Service Definition Language)
  - java2wsdl
- Create Web service Java code
  - Wsdl2java
- Implement server-side methods
  - org.simplegrid.util.SkeletonGen
- Packaging
  - Web service Java classes
  - SimpleGrid classes
  - Globus Java libraries
  - Web service description (WSDL and Axis2 service definition)
- Publish DMS Web service into Tomcat
- Develop Web service client
  - We will use the client within the SimpleGrid Web server to invoke the DMS application service



TeraGrid™

# Web Service Invocation

- **REST and SOAP**
  - Axis2 Web services support both REST and SOAP (since Axis2 1.4.1)
- **Programming APIs**
  - Java
    - Axis2-based client code
  - PHP
    - SOAP: nu-soap
    - REST: curl

TeraGrid™



TeraGrid™

# Gateway Computation Management

- **Gateway computation support workflow**
  - 1. Scientists give code to gateway admin
  - 2. Gateway admin compiles and deploys code to TeraGrid
  - 3. Gateway developer develops application and user interfaces
  - 4. Gateway admin publishes interface to scientists
  - 5. Scientists access application through gateway portal
- **SimpleGrid gateway computation management service**
  - Goal: help gateway admin and developer at step 3&4
  - A Web service implementation



TeraGrid™

# SimpleGrid Gateway Computation Management Service

- Service definition
  - [http://\\${tutorial\\_server}:8080/axis2/](http://${tutorial_server}:8080/axis2/)
- Computation on TeraGrid clusters: job wrapper
- Gateway-side computation management service
  - GWAppExecution Web service
    - Axis2-based Web service
    - Provides a generic interface for job submission, monitoring, and data transfer

TeraGrid™



TeraGrid™



# Part V: Sharing: SimpleGrid Gateway User Environment Development



**Application  
Gadgets**

**Web Server/  
Gadget Container**

**Development  
Gadgets**

Service invocation

**Gateway Services**

Grid-based computation

**TeraGrid**



**NCSA**



INDIANA UNIVERSITY

**SDSC**



TeraGrid™

# Technologies

- **Web 2.0 development**

- JavaScript

- AJAX

- Yahoo UI (YUI)

- Twitter API

- Google Map

- OGCE gadget container based on Open Social framework™

- **Web services**

- Axis2 Web service development

- GFAC/Xregistry service APIs

- SimpleGrid gateway computation management service API



TeraGrid™

# Web 2.0 Technologies

- **Web 2.0**

- A cloud

- Portlets, AJAX, JSON, REST, Wiki, blogging, social networking, semantic Web, ... ..

- Features

- Collaborative Web content management
  - Dynamic and flexible Web content presentation

- **Focus**

- YUI-based Web 2.0 user interface development

- Using Twitter to coordinate collaborative analysis



TeraGrid™

# User Interface: Rich Client vs Portal

## •Portal

- Portal is basically server-side technology for Web content management
  - MVC (Model-View-Control), the common portal/portlet development model, happens at server side.
  - Portal server controls Web content rendering
- Technology is well-developed

## •Web 2.0 rich-client user interface

- Rich client
  - Highly flexible and dynamic control of Web content presentation through JavaScript and Web browser DOM
  - Server role change: Web server and Web services are used for data-feeding and message-handling. The control of Web content rendering happens at client
- Technology is evolving



TeraGrid™

# User Interface: Rich Client vs Portal (Cont'd)

- **Impact on gateway user interface development**
  - Portal and rich-client user interface are complementary to each other
    - Server side control: user management, request routing, access control, etc.
    - Client side user-friendliness.
    - Existing TeraGrid gateways are adding more and more Web 2.0 features into gateway portals.
- **SimpleGrid**
  - SimpleGrid 0.7: Portal technology illustration
  - SimpleGrid 2.0: Web 2.0 user interface illustration



TeraGrid™

# Yahoo UI (YUI)

- Web 2.0 user interface technologies

- Yahoo UI (YUI)

- 4-year history
- Plenty of user interface components
- Quick learning curve
  - Plenty of online code examples

- jQuery

- YUI features

- User interface

- Sliders, menus, layouts, client-server communications, tables, ... ..

- Client-server communication

- AJAX
- Get utility for 3<sup>rd</sup>-party communication

- Event and data handling

- Drap and drop, timer, communications
- JSON, plain text, XML data support

- Development support

- Dynamic JavaScript library loading
- JavaScript library compressor



TeraGrid™

# Outline for SimpleGrid Gateway Development

- **Developing Web 2.0 user environment**

- Layout design
- Using YUI components to design a highly usable Web interface
- Using Google Map for analysis result visualization
- Using Twitter for collaboration
- Web service invocation

- **Gadget development**

- Integrating existing gateway applications into gadget container
- Implementing authentication among gadgets, container, and gadget services
- Inter-gadget communication



TeraGrid™

# SimpleGrid Gateway User Interface

- **Left pane: collaboration**

- Each gateway user participates a Twitter group for this tutorial
- Web server is the group messaging coordinator
  - Notify computation progress through Twitter updates

- **Middle pane: visualization**

- Google Map-based visualization on DMS results
- As results are available, users drag and drop a DMS job for visualization

- **Right pane: DMS computation management**

- Create and submit DMS jobs
- Automatic job status refresh
- Computation report



TeraGrid™



# Programming the SimpleGrid Gateway User Interface

- **Code location**

- \$HOME/simplegrid2/webapp

- **YUI programming**

- YUI libraries and stylesheets

- YUI data table programming

- DMS job list retrieval, sorting, and rendering

- DMS job selection, highlight, and drag&drop

- YUI dialog programming for DMS job submission

- Event handling for user actions and timers

- **Visualization**

- Invoking GISolve's Google Map visualization library



TeraGrid™

# SimpleGrid Gateway Server

- **Apache+PHP+MySQL-based Web server**

- Apache for HTTP request handling
- PHP for data feeding and Web service call invocation
- MySQL for permanent job data storage

- **Functions**

- Serve Web content data

- Index.html and other static Web resources (e.g., images)

- JavaScript libraries

- Handling DMS job creation, submission, refresh, listing, and deletion

- Invoking DMS Web service through Php Web service client

- Posting Twitter updates on job status change



TeraGrid™

# Science Gateway Web Portals

- **Conventional portals**
  - GridSphere, WebSphere
  - Liferay
  - Sakai
  - Apache Jetspeed
- **Web 2.0 portals**
  - OGCE gadget container

*TeraGrid*<sup>TM</sup>



TeraGrid™

# TeraGrid-based Computation Support in Gadgets

- **GFAC/XRegistry**
  - TeraGrid site management
  - Gateway application registration
  - Application service registration
  - Computation management
- **SimpleGrid gateway application development gadgets \***
  - GWApp development gadget
  - GWApp gadget
  - GWApp job gadget
- **TeraGrid gadgets**
  - MyProxy gadget

\*: under internal evaluation



TeraGrid™



DEMO: SimpleGrid as OGCE gadget

*TeraGrid*<sup>TM</sup>



TeraGrid<sup>TM</sup>

# Summary

- SimpleGrid provides basic components to develop science and engineering gateways
  - Abstraction of generic Grid middleware services
    - Hides the complexities of Grid computing
    - Enables science gateway developers to concentrate on developing PSE by working on reusable and extensible software components
  - Reusable components for building Web 2.0 gateway user environment
  - This tutorial includes code, links, and examples
  - Based on a real-world GIS application
- Gateway server setup instructions and additional materials are provided in GISolve/TeraGrid online documentation
  - TeraGrid science gateway wiki page
    - [http://www.teragridforum.org/mediawiki/index.php?title=Science\\_Gateway\\_SimpleGrid](http://www.teragridforum.org/mediawiki/index.php?title=Science_Gateway_SimpleGrid)



TeraGrid™

# Q & A

- Questions and comments?
- Thanks!

# TeraGrid™



TeraGrid™