Building Blocks for a Simple TeraGrid Science Gateway

-- A GI Solve Approach

Yan Liu and Shaowen Wang

Grid Research & educatiOn group @ ioWa (GROW)
The University of Iowa

June 4, 2007
Contributors/ Collaborators

- SDSC
  - Andrew Sanderson
  - Nancy Wilkins-Diehr

- UC/ANL
  - Stuart Martin

- UIowa
  - Eric Shook
Objectives

- Use TeraGrid to support domain-specific scientific computing
- Develop Grid-enabled applications to access TeraGrid capabilities
- Create a GridSphere-based portal as a TeraGrid science gateway interface
- Develop JSR-168 compliant portlets to build gateway components
- Understand a GISolve-based workflow to steer analyses on TeraGrid
Overview

Gateway Portal
- UserPortlet
- Job Submission
- Dataset Transfer
- Dataset Visualization

Grid-enabled Applications
- SimpleCred
- SimpleRun (GT2 & GT4)
- Simple-Transfer
- SimpleViz (Local)

TeraGrid Resources
- Security/Account
- Computing
- Storage
- Visualization

LEARN
Learning Curve

- Access TeraGrid resources
  - Accounts, computing and data storage
- Develop Java programs for TeraGrid access
  - JGlobus Cog programming
  - A simple visualization module
- Build a simple science gateway Grid portal
  - JSR-168 PortletAPI
  - GridSphere-based portlet development
Development Curve

- **Command line**
  - Running domain science application on TeraGrid

- **Grid-enabled application development**
  - Java programming
  - Application-specific SimpleGrid API

- **Science gateway**
  - SimpleGrid portal
  - A straightforward GIScience (Geographic Information Science) application workflow
Application Example

- **DMS Spatial Analysis**
  - Spatial interpolation based on the Dynamically Memorized Strategy (DMS)

- **Application package**
  - Binary executables
  - Sample datasets
  - Package directory layout

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

- TeraGrid access
  - Globus Toolkit 4.0.1 (CTSS3)
    - MyProxy, GRAM, WS-GRAM, GridFTP
- Programming
  - Java Cog kit
    - JGlbous module
- Grid portal
  - GridSphere portal server
  - GridSphere-based portlet development framework
  - JSP and GridSphere Visual UI
Environments

- Three tutorial portal servers
  - grow-dev1.its.uiowa.edu
- Portal server software
  - SSH client and server
  - Globus Toolkit 4.0.1 client tools
  - Java J2SE 1.5
  - Ant 1.6.5
  - Apache httpd 2.0.52
  - Tomcat 5.5.23
  - GridSphere 3.0.5 (customized)
  - MySQL server 4.0.20
- TeraGrid sites
  - UC, NCSA, and SDSC
User Accounts

- **TeraGrid training accounts**
  - train01 - train19
  - Gateway head nodes:
    - tg-login.ncsa.teragrid.org
    - tg-login.uc.teragrid.org (through ncsa or sdsc)
    - Tg-login.sdsc.teragrid.org

- **Portal accounts**
  - train1 - train 19
  - Password is dispatched with tutorial materials
  - URL: https://$\{portal_server\}/gridsphere/gridsphere
MySQL database accounts
- Same user names as TeraGrid training accounts
- Password same as your portal account password

Tutorial server UNIX accounts
- Same user names as TeraGrid training accounts
- Password dispatched with tutorial materials
- BASH shell environment

Tutorial package
- In $HOME/SimpleGrid
Exercise 1: Setup Development Environment

- Login to one of the three tutorial servers
  - SSH login: username@$tutorial_server

- Environment setup

  ```
  cd ~
  # add simplegrid-env.sh in your .bashrc
  vi simplegrid-env.sh
  ```
Outline

- Part I: Basic TeraGrid services and tools
- Part II: SimpleGrid API development for Grid-enabled analysis
- Part III: SimpleGrid portal development
Part I: Basic TeraGrid Services and Tools

- 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?
DMS Binary Package Deployment

- SSH login
- Setup your TeraGrid account environment
  
  ```
  ## 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
  ```

- Deploy DMS binary package to all three sites
  
  ```
  scp SimpleGrid/applications/dms.uc.tar.gz train1@tg-login.uc.teragrid.org:./
  ssh train1@tg-login.uc.teragrid.org:./
  tar xvfz ./dms.uc.tar.gz
  ```
**Get a Grid Proxy**

[tomcat@grow-dev1 ~]$ `myproxy-logon -l gisolve -t 100 -s myproxy.teragrid.org`

Enter MyProxy pass phrase:

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

[tomcat@grow-dev1 ~]$ `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/
Transfer a Sample Dataset

[tomcat@grow-dev1 test]$ globus-url-copy file:/home/tomcat/SimpleGrid/simplegrid/webapp/storage/samples/sample gsiftp://gridftp-hg.ncsa.teragrid.org:2811/~/sample1

[tomcat@grow-dev1 test]$ ssh gisolve@tg-login.ncsa.teragrid.org
tg-login2 ac/gisolve> ls -l sample1
-rw-r--r--  1 gisolve  lpt  1116491 2007-05-29 21:28 sample1

http://teragrid.org/userinfo/data/transfer_location.php

http://teragrid.org/userinfo/data/gridftp.php
GRAM Job Submission

- **RSL (for UC)**

```
&(jobType="single")
  (count=1)
  (host_count="1")
  (project=TG-SES070007N)
  (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 tg-grid.uc.teragrid.org:2120/jobmanager-pbs
globus_gram_client_callback_allow successful
GRAM Job submission successful
DONE
```
<job>
  <factoryEndpoint
    xmlns:gram="http://www.globus.org/namespaces/2004/10/gram/job"
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing">
  </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>
WS-GRAM Job Submission

- Command: `globusrun_ws -submit -f rsl_xml_file`

http://teragrid.org/userinfo/jobs/globus.php
Analysis Process

- Get a grid proxy
- Transfer a dataset to specified TeraGrid sites
- Submit Grid jobs to run DMS binary executables against the transferred dataset
- Collect results back
- Visualize results
  - Will be shown later
Part II: SimpleGrid API Development for Grid-enabled Analysis

- How to do Part I in a programming way?
- Purpose
  - Automate the access to TeraGrid resources as a Grid-enabled application
- Package
  - Location: SimpleGrid/simplegrid/src/
  - SimpleGrid API for DMS analysis
    - org.gisolve.demo.app.*
    - org.gisolve.demo.grid.*
    - org.gisolve.demo.util.*
  - Libraries
    - All jars from ${GLOBUS_LOCATION}/lib
      - Particularly cog-jglobus.jar
    - JFreeChart libraries for visualization

Get a Grid Proxy

- Class
  - org.gisolve.demo.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()

- Export proxy to a file
  - SimpleCred::export()
Transfer a Sample Dataset

- Class
  - `org.gisolve.demo.grid.data.SimpleTransfer`

- Local $\Rightarrow$ remote transfer
  - `SimpleTransfer::remote2local()`
  - `SimpleTransfer::local2remote()`

- Third-party transfer
  - Refer to Cog manual (version 1.1)

http://wiki.cogkit.org
GRAM Job Submission

- **Class**
  - org.gisolve.demo.grid.job.SimpleRunGT2
  - org.gisolve.demo.grid.job.SimpleRSL

- **RSL generation**
  - Application specific

- **GT2 job submission**
  - Use batch mode in which a call returns immediately with a job handle
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
WS-GRAM Job Submission

- **Class**
  - org.gisolve.demo.grid.job.SimpleRunGT4
  - org.gisolve.demo.grid.job.SimpleRSL

- **RSL generation**
  - Application specific

- **GT4 job submission**
  - Use batch mode in which a call returns immediately with a job handle
  - Note that it uses a different globus package org.globus.exec.*

http://www.globus.org/toolkit/docs/4.0/execution/wsgram/developer-index.html#id2563059
How to Write RSL?

- RSL schema
  - http://globus.org/toolkit/docs/4.0/execution/wsgram/schemas/gram_job_description.html

- GT2 and GT4 schema comparison
Exercise 2

- Test SimpleGrid API
  - Source simplegrid-env.sh
  - SimpleGrid/bin/runTest2.sh
Part III: SimpleGrid
Portal Development

- Science gateways to TeraGrid
  - Provide an online access points to TeraGrid
  - Aggregate domain science application-level capabilities
  - Hide the complexity of using TeraGrid
  - Portal is commonly used to meet such needs

- Focus
  - Provide transparent access to TeraGrid for domain scientists
  - Build application-oriented workflow inside portal for easy access by users
  - Technical details for portlet development
    - GridSphere
    - PortletAPI
    - GridSphere-based portlet development
GridSphere

■ Features
  - http://www.gridsphere.org
  - Open source
  - PortletAPI (JSR-168) compliant
  - Object persistence support through Hibernate

■ GridSphere 3.0.5
  - Download: https://svn.gridsphere.org/
  - Requirements: java, ant, tomcat, mysql
  - Build: ant install
    ▪ Be sure to change hibernate.properties, build.properties
    ▪ Copy mysql jdbc driver to ${CATALINE_HOME}/common/lib/
  - Development using Eclipse
    ▪ Instructions will be added in our online documentation
Exercise 3

- Demo
  - GridSphere interface
  - New portlet project creation
SimpleGrid Portlets

- Location: SimpleGrid/simplegrid
- Two portlets
  - UserPortlet
    - Display user information
    - Initialize or renew grid proxy
      - Automatic renewal after initialization
  - DMSPortlet
    - DMS workflow control
    - Build user interfaces for DMS analysis
- Each tutorial user will build and deploy their version of SimpleGrid portlets (project name is simplegrid_${USER})
Introduction to SimpleGrid Development

```java
package org.gisolve.demoportlet;

// GridSphere Visual UI
import java.text.SimpleDateFormat;
import net.sf.jsfgrid.util.

public class UserPortlet extends ActionPortlet {

    public void init(PortletConfig config) throws PortletException {
        super.init(config);
        DEFAULT_VIEW_PAGE = "show";
    }

    // Other methods...

    public void show(RenderFormEvent event) throws PortletException {
        PortletSession mysession = req.getPortletSession(true);
        // Get grid information
        User user = (User) req.getAttribute(ServletRequest.ATTRIBUTE_PORTLET_USER);
        HttpServletRequest request = req.getHttpRequest();
        String userId = user.getUserId();
        try {
            UserBean userBean = new UserBean(userId);
            // Initialize the userBean instance
        } catch (IOException ioe) {
            throw new PortletException("Error loading SimpleGrid configuration file: " + configPath,
                                          PortletRequest.ERROR_PAGE,
                                          PortletResponse.IN_VIOLATION
            );
        }

        // Get configuration information
        String configPath = this.getPortletContext().getRealPath("simplegrid.properties");
        Config myconfig = null;
        // Other code...
        try {
            String server = myconfig.getString("grid.myproxy.server");
            // Other code...
        }
    }
}
```
SimpleGrid Project Layout

- src/org.gisolve.demo.portlets
  - Portlet action classes
- lib/
  - Libraries for Globus and JFreeChart
- webapp/jsp/
  - JSP pages for user interface rendering
- webapp/WEB-INF/persistence/hibernate.properties
  - Configuration for backend database access
- webapp/WEB-INF/portlet.xml
  - Portlet definition
    - Defines unique portlet id, the portlet class, etc.
- webapp/simplegrid.properties
  - SimpleGrid specific configuration about Grid resources
  - Loaded at the first time the UserPortlet is visited
- build.xml
  - Ant build file to compile, deploy, and install SimpleGrid portlets to GridSphere
  - Need to change if you have your own libraries or packages developed
- build.properties
  - Define project name and location of GridSphere where SimpleGrid finds dependent GridSphere java classes and PortletAPI (javax.portlet.*) definitions
Runtime Configurations

- simplegrid.properties
Portlet Container

- PortletAPI (J SR-168 specification)
  - Portlet specification
  - What are not defined?
    - Object persistence: store portlet status in permanent storage
    - Portlet rendering method
      - JSP, Velocity, GridSphere Visual UI (JSP taglib), Java Server Faces

- GridSphere: J SR-168 compliant portlet container
  - Implemented javax.portlet.*
  - Hibernate as object persistence technologies
  - JPS-based Visual UI rendering
  - What are not standardized?
    - ActionPortlet based on Visual UI
    - Portlet service framework (from WebSphere)
  - Other open source portlet containers
    - Apache Jetspeed/Pluto, Sakai
PortletAPI (JSR-168 Specification)

- Sample portlet
  - GridSphere jsrtutorial: ActionHelloworld

- Portlet modes
  - VIEW (doView()): the default display mode
  - EDIT (doEdit()): preference/configuration editing
  - HELP (doHelp()): help information

- Base portlet: GenericPortlet

- PorletURL
  - Generate actionURL and renderURL in response page

http://www.gridsphere.org/gridsphere/gridsphere/guest/download/r/
PortletAPI (JSR-168 Specification)

- PortletRequest
  - Get parameters from portlet container
    - ActionRequest
      - Used in action processing
    - RenderRequest
      - Used in rendering
  
- PortletResponse
  - Pass parameters to portlet container
    - ActionResponse
      - Parameters set in ActionResponse will be passed to RenderRequest
    - RenderResponse
      - Used in rendering response page

- Action processing
  - processAction(ActionRequest, ActionResponse)

- Default rendering
  - doView(RenderRequest, RenderResponse)
Portlet definition (portlet.xml)

```xml
<portlet>
    <description xml:lang="en">SimpleGrid user portlet</description>
    <portlet-name>SimpleGridUserGisolve</portlet-name>
    <display-name xml:lang="en">SimpleGrid user gisolve</display-name>
    <portlet-class>org.gisolve.demo.portlets.UserPortlet</portlet-class>
    <expiration-cache>0</expiration-cache>
    <init-param>
        <name>aname</name>
        <value>avalue</value>
    </init-param>
    <supports>
        <mime-type>text/html</mime-type>
        <portlet-mode>view</portlet-mode>
        <portlet-mode>edit</portlet-mode>
        <portlet-mode>help</portlet-mode>
    </supports>
    <supported-locale>en</supported-locale>
    <portlet-info>
        <title>SimpleGrid User home for gisolve</title>
        <short-title>simplegrid-gisolve</short-title>
        <keywords>simplegrid, gisolve, user</keywords>
    </portlet-info>
    <portlet-preferences>
        <preference>
            .......
        </preference>
    </portlet-preferences>
</portlet>
```
Portlet Development

Questions
- How to direct actions to corresponding action methods?
  - Avoid a big processAction()
- How to get parameters from PortletRequest?
- How to render response pages?
- How to communicate between portlets?
Portlet Development

- GridSphere Visual UI
  - Rendering: JSP tag-lib
  - Action: ActionPortlet

- Velocity
  - Rendering: Velocity templates
  - Action: VelocityPortlet

http://www.gridsphere.org/gridsphere/docs/TagGuide/TagGuide.html
http://www.collab-ogce.org/ogce2/velocity-portlets.html
Exercise 4

- **Deploy your SimpleGrid project** (simplegrid_${USER}) to gridsphere container
  - Use “ant install” for a fresh install
  - Use “ant deploy” for update (it does not clear database content)
  - Look at $CATALINA_HOME/webapps/simplegrid_${USER}/WEB-INF/lib to see project jars generated by deployment

- **How does Gridsphere know your project?**
GridSphere-based Portlet Development

- Visual UI taglib for rendering
  ```jsp
  <%@ taglib uri="/portletUI" prefix="ui" %>
  <%@ taglib uri="http://java.sun.com/portlet" prefix="portlet" %>
  <portlet:defineObjects/>
  ```

- Portlet class
  - Extend ActionPortlet
  - All beans are constructed from ActionFormEvent or RenderFromEvent
  - Get PortletRequest and PortletResponse from FormEvent
  - Use `setNextState()` to point to JSP page in rendering methods
  - Use `setNextState()` to point to rendering method in action methods
Example: UserPortlet

- **UserPortlet::show()**
  - Default rendering method which displays user information and grid proxy configuration in JSP
  - In JSP, Bean is used to render the values of user forms
    - Java code is embedded to directly access RenderRequest for grid proxy information

- **UserPortlet::configSimpleCred()**
  - Action method to retrieve grid proxy either from MyProxy server or locally from a valid proxy file
Velocity-based Portlet Development

- Portlet definition
  - Initial template name
  - velocity.properties

- Action class
  - Get parameters: ActionRequest.getParameter()
  - Rendering: VelocityContext.put(name, object)
  - Action methods: called based on their names through Java Reflection

- Velocity template

```html
#if( !$gisolve_portal_error_info.equals("") )
   <pre><p><font face="Arial Narrow" color="red">$gisolve_portal_error_info</font></p></pre>
<hr>
#end
<form name="dms0002" method="POST" action="$actionURL">
   <input type="submit" value="Return" name="actionMethod_doDms_return">
</form>
</p>
```

http://velocity.apache.org/engine/releases/velocity-1.5/user-guide.html
DMS Analysis Workflow in Portal

- Workflow control is done by keeping state information in PortletSession as attributes

- Portlet communication
  - Need to coordinate two portlets
    - PortletSession.getAttribute(name, PortletSession.APPLICATION_SCOPE)
    - PortletSession.setAttribute(name, value, PortletSession.APPLICATION_SCOPE)

- Learn how SimpleGrid remembers current status of each job
DMSPortlet Interface

Create a DMS job:

Dataset:  
K-neighbor: 20  
Resolution (NxN): 500

Grid site selection: UC SDSC NCSA

Done
Exercise 5

- Run DMS portlet and get visualization results
  - Result page plots results after the analysis
  - Do you see your job is running on TeraGrid?
    - Refresh job status
    - Verify it by using “qstat | grep gisolve” to see the status of your job

- Thread in SimpleGrid portlets
  - File transfer and result visualization access the same portal server and are time-consuming
  - Threads are used to respond to user requests quickly
Visualization on TeraGrid

- SimpleGrid uses JFreeChart for simple local visualization
- We are evaluating the integration between GISolve and ParaView on TeraGrid to support advanced visualization

http://tg-portal.uc.teragrid.org/
Summary

- SimpleGrid provides basic components to develop science and engineering gateways
  - This tutorial includes code, links, and examples
  - Based on a real-world GIS application
- SimpleGrid is also used in GISolve, TeraGrid GIScience gateway
- Gateway server setup instructions and additional materials will be provided in GISolve/TeraGrid online documentation
Q & A

- Questions and comments?
- Thanks!