Portal Technology

Grid Computing (M)

Tutorial 8

Jipu Jiang,

jiangj@dcs.gla.ac.uk

UNIVERSITY *of* GLASGOW



Introduction (to me!)

- My name is Jipu Jiang, M.Sc. Advanced Computing Science from University of Glasgow. I am currently working on the ESP-GRID project in the National e-Science Centre (NeSC).
- **ESP-GRID** is a project with Oxford university and NeSC of University of Glasgow.
  - **Aim:** To achieve a deeper understanding of the potential role that Shibboleth can play in grid authentication, authorization and security.
  - **Shibboleth**
    - **An architecture** that enables organizations to build single sign-on environments that allow users to access geographically distributed web-based resources using a single login.
    - It also defines **a secure way of exchanging information** between an organization and a provider of digital resources (such as data, video, documents, and so on).
    - Can be used for both **Authentication** and **Authorization**
    - Powerful architecture for building dynamic **Virtual Organizations** (VOs)
- The next project I am going to undertaken is **GLASS** (GLASgow university early adoption of Shibboleth) led by Prof. Richard Sinnott.
  - Aim: explore the usage and adoption of Shibboleth in both education and medical domain including those requiring Grid-based authentication and authorization.

Copyright © 2006 University of Glasgow

Outline of this tutorial

- Portals
  - What is Portal?
  - What is a Grid Portal?
  - Why do we need Grid Portals?
- Grid Portal Frameworks
  - GridSphere vs WebSphere
- Portlets
  - Portlets vs Servlets
  - Portlet Standards
  - Portlet Development Process
  - Grid Portlets
  - Portals in Grid Projects
- How far GridSphere fills the need of Grid community?
- Outlook for the future
- Demonstration
- Useful resources for more information on portals and portal technologies

Copyright © 2006 University of Glasgow

What is Portal?

- A web portal is a base that acts as a starting point to help the navigation through huge amounts of information on the World Wide Web.
  - Such as Yahoo!
- A web portal is different from a web search engine, such as Google.
  - Search engines tell us where the information is, it does not aggregate it.
  - Instead, a web portal aggregates information and provide index services.



Copyright © 2006 University of Glasgow

Grid Portal

- Grid Portals build upon the familiar Web portal model to deliver the benefits of Grid computing to virtual communities of users, providing a single access point to Grid services and resources.

- The major difference is that a Grid Portal is for Grid resources rather than classified web pages/sites.

- A Grid portal is a web based application that commonly provides personalization, single sign on, content aggregation, and seamless access to Grid heterogeneous resources and services that support the end user in one or more tasks through a Web-based user interface.

Copyright © 2006 University of Glasgow

Why do we need Grid Portals?

- How to invoke a Grid Service?
- Take the MathService in “Writing Your First Grid Service in 5 Simple Steps ” as an example:
  - Write up your own Client.java,
  - Configure environment variables by running:

    %> source $GLOBUS_LOCATION/etc/globus-devel-env.sh
  - Compile the client:

    **%> javac -classpath ./build/classes/:$CLASSPATH **
    org/globus/progtutorial/clients/MathService/Client.java
  - Run:

    **%> java -classpath ./build/classes/:$CLASSPATH **
    **org.globus.progtutorial.clients.MathService.Client **
    http://127.0.0.1:8080/ogsa/services/
    **progtutorial/core/first/MathService **
    5

Copyright © 2006 University of Glasgow

Why do we need Grid Portals …ctd

- This is not too difficult – although will often put off non-IT savvy researchers
- But what if
  - more many parameters are needed.
  - a more complicated CLASSPATH need to be set, e.g. necessary classes in various locations.
  - parameters must be given by the right order,
  - the parameter format must be correct, e.g. 12.34, not 12.3
  - Like this (GridBlast Service, Client):
    - %> java -classpath ./build/classes/:.build/lib/:.lib/:\
      $CATALINA_HOME/shared/lib/:$CATALINA_HOME/common/lib/:\
      $CATALINA_HOME/common/endorsed/:$CLASSPATH \
      bridges.jobsub.portlets. TestBridgesClient \
      http://127.0.0.1:8080/ogsa/services/bridges/JobServiceFactoryService \
      michab, Job1, >seq1\nATCTAGTACTAGTACTGTACTGATCA\n\
      >seq2\nACCATTTGATACACGATTAGCAATGA\n\
      >seq3\nTCGTAGATAGATGATTGATGATGTGA\n, \
      -p blastn, ecoli.nt, 10, 7, yes, no, jiangj@dcs.gla.ac.uk, html

- These situations may lead to complex lists of error messages, and it is often hard to find out where the errors are.

Copyright © 2006 University of Glasgow

Command Line Interfaces

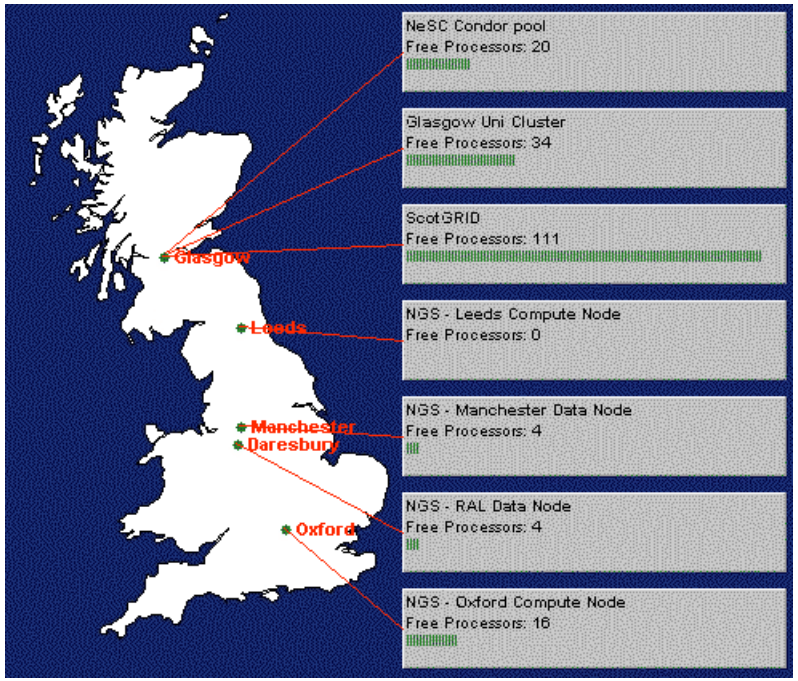
- May be not a problem for computer scientist.
- But “IS” for non-computer scientists and suffers from
  - takes time for users to get used to command line options.
  - technologies often are too complex to be used by scientists who may lack the expertise needed for dealing with such a complex infrastructure
    - they will likely need client side Grid installations
  - Error-prone, e.g. type a letter wrong in the Grid Service URI, forget a parameter etc
  - It becomes more difficult if more than one Grid Service is running simultaneously
    - Which service, how to invoke, makes client side generally harder
  - Difficult to integrate or interact with multiple Grid Services
  - Limited response makes it hard to visualize the results, e.g. when lots of data is returned to the command line how should it be interpreted
- These are problems of command line interface in Grid Computing.

Copyright © 2006 University of Glasgow

Portal Interface

- User interfaces in Grid Computing have tended to follow either a command-line or a portal-based approach
  - Command line interfaces, such as provided by the Globus Toolkit, offer a flexible and powerful but user-oriented environment.
  - Grid Portals deliver their functionality through Web browsers or PDAs with friendlier user interfaces to restricted sets of services.

- The Web-based interfaces are more familiar to non-IT savvy users, and graphic interfaces can be used to display more sophisticated information to researchers.
- The future Grid should be as easy to use as the internet!!!



Copyright © 2006 University of Glasgow

Benefits of Grid Portals

- Grid Portals can hide the complexity from users
- A Grid portal provides end users with a customized view of software and hardware resources specific to their particular problem domain.
- They also provides a single point of access to Grid-based resources and allow a single source through which authorization can be checked/enforced.
- They allow scientists or engineers to focus on their problem area by making the Grid a transparent extension of their desktop computing environment.
- Grid Portals facilitate the establishment of VOs, comprised of distributed researchers allowing them to collaborate and access heterogeneous resources and services more efficiently and seamlessly

Copyright © 2006 University of Glasgow

Grid Portal Frameworks

- Previous Grid portal developments show up several problems.
  - they are generally hard coded with underlying Grid infrastructure details and the associated code base.
  - the lack of real usability has made it difficult to test and evaluate user interfaces.
  - web application development still remains a tedious task with little in the way of reusable components, forcing developers to constantly ”re-invent” the wheel.
- To solve those problems, Grid Portal Frameworks were developed to support rapid establishment of portals and improve component reuse.
- Two commonly used portal server packages
  - **GridSphere** Portal Framework - An open source research project from the EU GridLab project www.gridsphere.org
  - **WebSphere** Portal - commercial software solution from IBM http://www-306.ibm.com/software/websphere/

Copyright © 2006 University of Glasgow

GridSphere vs WebSphere

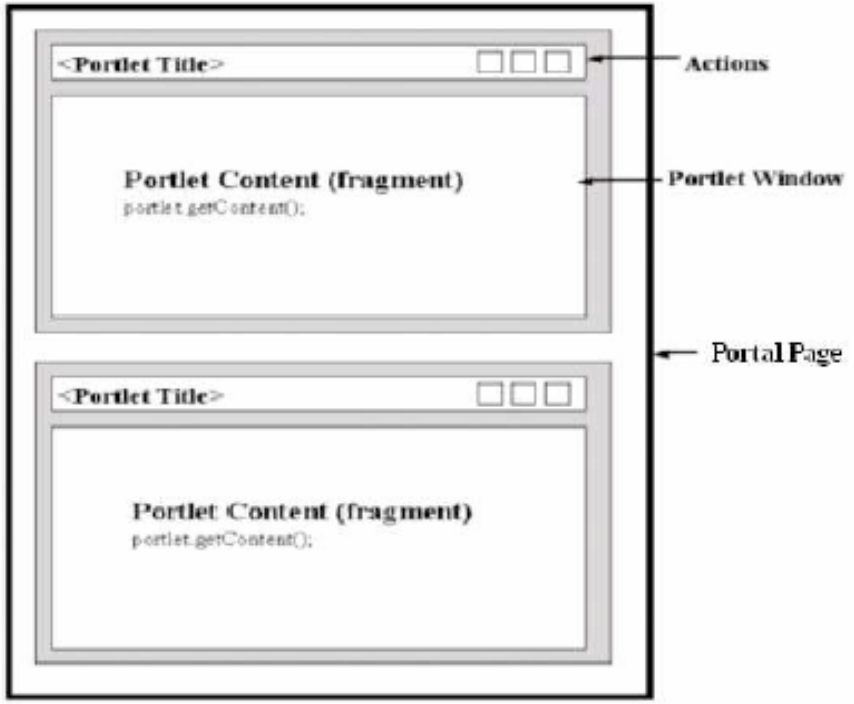
Feature	*Gridsphere*	*Websphere*
License	Open Source	Commercial Product
Complexity	Low	High
Re-use of Publicly Available Components	Good – deploys into Apache Tomcat and uses log4j	Poor – proprietary implementations of application server, logging package and http server
Documentation	Good – about the right amount	Almost too much – things are hard to find
Installation/Configuration	Easy & Quick	Cumbersome – has to be downloaded as lots of separate files
Special Hardware Requirements	No	At least 1 GB memory per processor and At least 4GB disk space
Debugging Your Own Portlets	Easy – log4j can be used	Hard – in-code debugging statements need to be changed to support IBM logging mechanism
Stable	Not very stable	Better than GridSphere
JSR168 Compliance	Full	Full from version 5.1 onwards
WSRP Compliance	No	Yes
Security	No many security features	Very strong, even URLs are encrypted
Overall Impression	Lightweight, easy-to-use, basic	Complex, monolithic, feature-rich

Copyright © 2006 University of Glasgow

Portlets

- Both GridSphere and WebSphere are built on portlets
  - Portlets are the building blocks of portals.
  - They look like little windows inside a typical Web page.
  - Portlets act as pluggable user interface components.

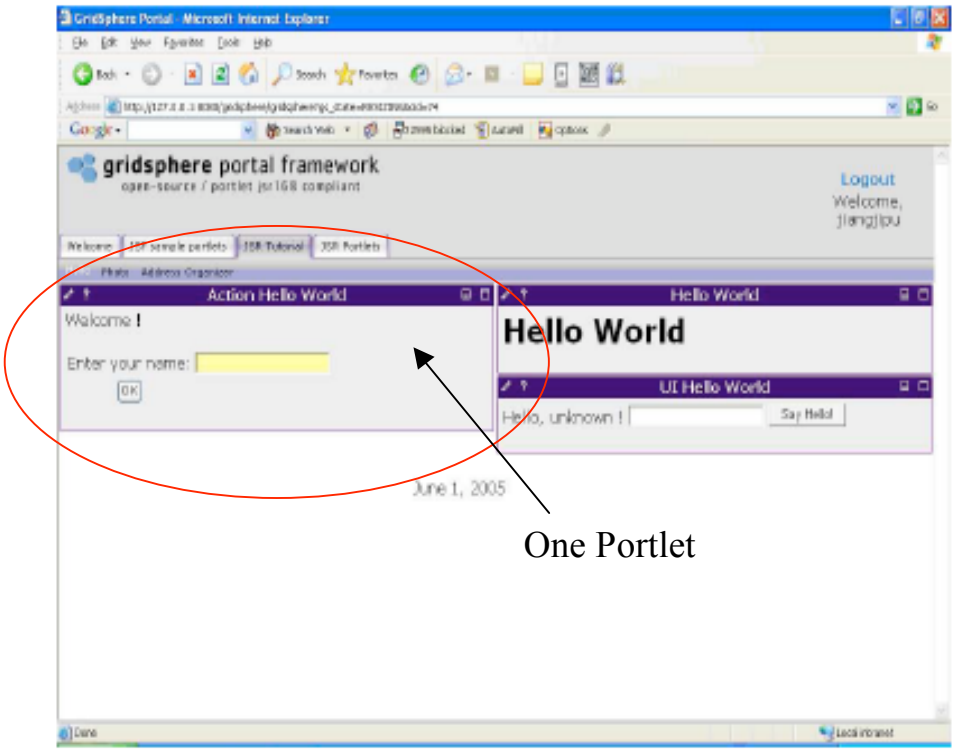
- Two perspectives of Portlets:
  - From a users' perspective: A window in a portal that provides a specific service
  - From an application development perspective: A portlet is a software component written in Java, managed by a portlet container, which handles user requests and generates dynamic contents.



Copyright © 2006 University of Glasgow

Portlets vs Servlets

- Portlets are very similar to Java Servlets. They both process HTTP requests and produce HTML output, e.g. with JSP
- But
  - Portlets are a specialized and more advanced form of Java Servlets.
  - Portlets run in a portlet container inside of a Servlet container (e.g. Tomcat, Axis etc.), which is a layer that runs on top of an application server.
  - Portlets' HTML output is only a small part of a Web page. The portal server fills in the rest of the page with headers, footers, menus, and other portlets.



Copyright © 2006 University of Glasgow

Portlets Standards

- Standards for Portlets
  - **JSR-168** (Java Portlet Specification) defines a standard Portlet API that is specific to Java-based portals.
    - This standard attempts to ensure the compatibility of portlets developed under different portal frameworks.
  - **WSRP** (Web Service for Remote Portlets) defines a universal API that allows portals of any type to consume portlets of any type.
    - This standard attempts to ensure that remote portlets can be invoked
- WSRP and JSR-168 are complementary specifications
  - JSR-168 builds the foundation for WSRP and provides the possibility of inter-operability of portlets
- These two standards have becoming industry standards, and are widely adopted.
- However JSR-168 is not perfect
  - tweaking is often needed to drop a JSR-168 portlet to a different portal containers
  - portal development frameworks typically provide their own “improved” APIs based on JSR-168.

Copyright © 2006 University of Glasgow

Portlet Development Process (GridSphere)

- GridSphere team developed a set of Ant scripts to ease the developing process.
- Typically, the basic programming process is made up by 7 steps:
  1. Create a new Gridsphere project: create the empty portlet by running the Ant script ”ant new-project”. Then a new folder with the name that you entered will be created under the <gridsphere_root>/projects/
  2. Write the Java code and place the portlet java source code in the <portlet_root>/src/
  3. Place your JSP (Java Server Page) files in <portlet_root>/webapp/jsp
  4. Edit the 3 XML configuration files located under <portlet_root>/webapp/WEB-INF/
  5. Run ”ant install” under <portlet_root> to compile the source
  6. Run ”ant deploy” under <gridsphere_root> to deploy the project into tomcat
  7. Restart the Tomcat container

Copyright © 2006 University of Glasgow

HelloWorld Portlet (JSR-168)

```java
package nesc.demo.gridsphere.helloworld;

import javax.portlet.*;
import java.io.IOException;

/**
 * A classic helloWorld portlet with some action
 *
 */
public class HelloWorldPortlet extends GenericPortlet {
    public void processAction(ActionRequest req, ActionResponse res) throws PortletException, IOExcep

        String name = req.getParameter("name");

        // must be passed into render parameters
        if (name != null) res.setRenderParameter("name", name);

    }

    public void doView(RenderRequest request, RenderResponse response) throws PortletException, IOExc

        // retrieve the name
        String name = request.getParameter("name");
        if (name != null) request.setAttribute("name", name);

        // create action url
        PortletURL url = response.createActionURL();
        request.setAttribute("url", url.toString());

        // create action url
        PortletURL rurl = response.createRenderURL();
        rurl.setPortletMode(PortletMode.EDIT);
        request.setAttribute("rurl", rurl.toString());

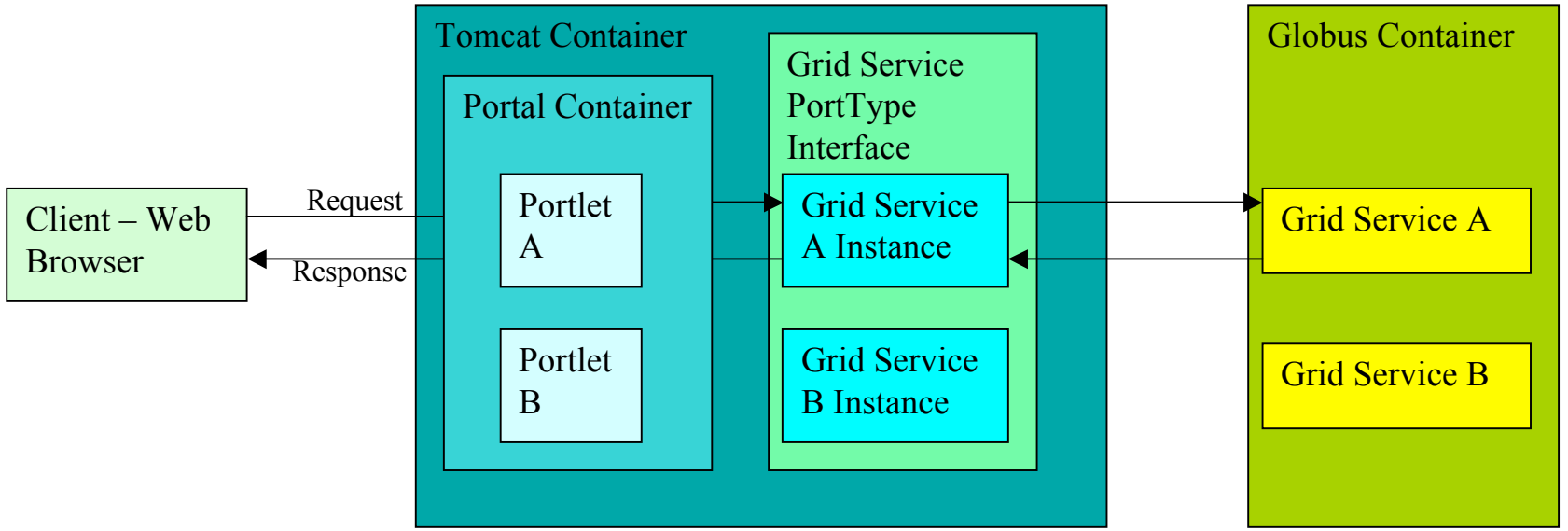
        getPortletConfig().getPortletContext().getRequestDispatcher("/jsp/helloworld/actionhello.jsp"
    }

}
```

Copyright © 2006 University of Glasgow

Grid Portlets

- Almost the same as normal Portlets programming, but with a few differences.
- Grid Portlet architecture
  - Users interact with a portlet inside a Portal
  - The portlet interacts with the Grid service inside the Globus container
  - Passing parameters, and get results back
  - Display results in the portlet



Copyright © 2006 University of Glasgow

Grid Portlets

- Preparation before programming.
  - Must have OGSA library in Tomcat.
    - Install the GridPortlet, which is a set of portlets developed by the GridSphere people as a plug-in to support the Grid. GridPortlets will deploy necessary OGSA library into Tomcat.
  - Must have Grid Service library in <portlet_root>/lib
    - Also need to have axis.jar and ogsa.jar there.
- Java coding (Grid Search Sort Service as example)
  - Define URI of the service
  - Invoke the Grid service by adding these lines:

```
101     try {
102         URL GSH = new URL(url);
103         SearchSortGridServiceGridLocator ssServiceLocator = new SearchSortGridServiceGridLocator();
104         searchSort = ssServiceLocator.getsearchSortGridServicePort(GSH);
105         numbOccurancies = searchSort.searchMethod(searchTerm, inputFile.getAbsolutePath());
106     } catch (Exception e) {
107         System.out.println("Error connecting to search service.");
108         e.printStackTrace();
109       }
```

Copyright © 2006 University of Glasgow

Portals in Grid Projects

- ESP-GRID explores the potential way of integration of Shibboleth and Grid through portals.
- Shibboleth framework can be used to support both Authentication and Authorization.
- What we have done in Glasgow is to setup a Shibboleth environment and to integrate it with a Grid Portal which links to some Grid Services
  - (There will be more details on Shibboleth in later labs)
- GridSphere was chosen as the Grid Portal Framework since
  - Lightweight framework
  - White-box framework, open source, easy to modify the code
  - Well support for Grid/Globus, by installing GridPortlets plug-in
  - Compatible to most academic technology, such as Log4j, Apache, Tomcat, etc.
  - Well integrated with Apache Server through mod_jk.

Copyright © 2006 University of Glasgow

Details in ESP-GRID

- Get rid of the authentication module of GridSphere.
  - GridSphere has its own authentication modules. To make authentication through Shibboleth, we have to remove such functionality from GridSphere.
- Redirecting HTTP requests from Apache to Tomcat
  - Shibboleth is an Apache module.
  - And GridSphere container is in Tomcat
  - To combine these two, mod_jk is used to make sure all the HTTP requests go to Apache first, then redirect them to the Grid Portal container in Tomcat.
- Retrieval roles associated with the user, and pass to Portal
  - Shibboleth can pass user information securely. Release of user information are controlled by the policy specified in Shibboleth. User information will be passed to the Portal hiding inside the HTTP headers.
  - Portlets inside Grid Portal must read this user information from headers, and make authorization decisions based on that.
  - Will be shown in demo.

Copyright © 2006 University of Glasgow

Does GridSphere fill the Grid community needs?

- GridSphere Portal framework offers many advantages over other portal frameworks. It is lightweight, small, easy to customize, and facilitates rapid portlet programming and deployment etc.
- But it has some insufficiency for the Grid community
  - It does not support features for searching for portlets
  - It does not support Grid Certificates and typicvally requires plug-ins for MyProxy – an online certificate repository
  - It is not well integrated with GT4 – still has a lot issues with WSRF, but has better support for GT3.2+
  - Poor Authorization. Though GridSphere supports Role Based Access Control (RBAC), such functionality is very primitive. Only 4 roles are allowed in the portal. The newest GridSphere release improves RBAC by allowing self defined roles, which is a good start.
  - Poor support for other authorization mechanisms such as PERMIS. Although the GridSphere Portal is an open source framework. It is not flexible enough right now and has been targeted explicitly to Globus but other Grid middleware exists also.
- Although GridSphere has a long way to go right now, considering it's a young product, it is still not bad. Given time, I believe it will be a very powerful portal framework for the Grid community in the future – once the rest of Grid technology stabilises!!!

Copyright © 2006 University of Glasgow

Outlook for the future

- Although elementary Grid services exist that enable scientific application developers to authenticate, access, manage, and discover sophisticated remote resources, these frameworks are not compatible with the commodity technologies and frameworks used by application scientists today.

- Portal technology can be the bridge connecting Grid with other researchers (i.e. the majority of people).

- With Grid Portal technology, accessing the Grid is no longer restricted to computers. PDAs, mobile phones, and other portable devices will also be able to enjoy the power of Grid technology through the web based access to portals.

Copyright © 2006 University of Glasgow

Demo

- Grid Search Sort Portlet integrated with Shibboleth
- Grid Blast Service integrated with Shibboleth.
- In both cases, Shibboleth is in charge of both authentication and authorization.

Copyright © 2006 University of Glasgow

Useful resources

- GridSphere web site
  - www.gridsphere.org
- GridSphere Progrmming
  - http://www.nesc.ac.uk/talks/549/Day_1_1215_wehrens.pdf
- JSR-000168 Portlet Specification
  - http://www.jcp.org/aboutJava/communityprocess/review/jsr168/
- Other portal frameworks
  - IBM WebSphere
    - http://www-306.ibm.com/software/genservers/portal/
  - JetSpeed v2.0
    - http://portals.apache.org/jetspeed-2
  - Jportlet
    - http://jportlet.sourceforge.net/
  - uPortal
    - http://www.uportal.org
  - LifeRay
    - http://www.liferay.com
  - Oracle Portal
    - http://www.oracle.com/technology/products/ias/portal/index.html
  - BEA Weblogic
    - http://www.bea.com

Copyright © 2006 University of Glasgow