

Create a grid testbed portal using Gridscape

Presented by developerWorks, your source for great tutorials

ibm.com/developerWorks

Table of contents

If you're viewing this document online, you can click any of the topics below to link directly to that section.

1. Before you start	2
2. Introducing Gridscape and Globus MDS	5
3. Design and implementation	8
4. Case study	12
5. Summary	27

Section 1. Before you start

What is this tutorial about?

When you're setting up a grid, you need a testbed to monitor the status of your resources, but it can take a substantial amount of time and effort to develop a custom grid testbed portal from scratch. Gridscape is a tool that aims to solve this problem by enabling the rapid creation of interactive and dynamic testbed portals without the need for any programming effort. The current version of Gridscape works with Globus 2.4, but work is underway to create versions that work with Globus 3.2 and the upcoming Globus 4.

In this tutorial, we'll show you how to set up and use Gridscape. We'll discuss:

- The Gridscape architecture
- How Gridscape interacts with Globus 2.4
- How to download and install Gridscape
- How to create a personalized grid testbed portal

Should I take this tutorial?

The focus of this tutorial is to help you get familiar with Gridscape and guide you through creating your first testbed portal. The tutorial works through an example and explains each step as you go along. You should take this tutorial if you are thinking of creating a personalized grid testbed portal.

The tutorial uses an example based on the Global Data-Intensive Grid Collaboration which was established by the Gridbus Project of the University of Melbourne. The collaboration was nominated as one of the finalists for the HPC Challenge event organized as a part of the IEEE/ACM Supercomputing Conference (SC 2003) held at Phoenix, Arizona, USA, in November 2003. The collaboration put together a World-Wide Grid (WWG) that contains over 200 grid nodes (PCs, workstations, clusters, supercomputers, databases, and applications) contributed by organizations and volunteers based in Australia, Asia, Europe, North America, and South America.

Prerequisites

Before proceeding further into the tutorial, take a look at several prerequisites that you need to satisfy:

- You should already know grid basics, have some exposure to the Globus 2.4 system, and maybe even have your own testbed that you can model through Gridscape. No programming effort is required, however.
- You should also have Java technology installed on your system. If not, you can download the latest release of the Java Development Kit (J2SE SDK) from [Sun Microsystems](http://java.sun.com/) (<http://java.sun.com/>) (<http://java.sun.com/>). (We recommend that you use release version 1.4 or later.) Install it according to the instructions provided.
- You need Jakarta Tomcat. Download the latest release from [The Apache Jakarta Project](http://jakarta.apache.org/tomcat/index.html) (<http://jakarta.apache.org/tomcat/index.html>) (<http://jakarta.apache.org/tomcat/index.html>). Gridscape was developed with release 4.1, and so it is recommended that you use this release as well to ensure that problems are kept to a minimum. Install it according to the instructions provided.
- Jakarta STRUTS is also required. Download the latest release of Jakarta STRUTS from [The Apache Jakarta Project](http://jakarta.apache.org/struts/index.html) (<http://jakarta.apache.org/struts/index.html>) (<http://jakarta.apache.org/struts/index.html>). Gridscape was developed with release 1.1.0, so this is the recommended version to use. Install it according to the instructions provided. To make things simpler, the STRUTS class and jar files have been included with the Gridscape package.

About the authors

Choon Hoong Ding is working as a Research Assistant in the Grid Computing and Distributed Systems (GRIDS) Laboratory at the University of Melbourne, Australia. He completed a Bachelor of Information Systems degree at Multimedia University, Malaysia, and a Master of Software Systems Engineering from the University of Melbourne, Australia.

Hussein Gibbins worked as a Research Assistant in the Grid Computing and Distributed Systems (GRIDS) Laboratory at the University of Melbourne, Australia. He completed a Bachelor of Software Engineering degree from the University of Melbourne, Australia.

Rajkumar Buyya is a Senior Lecturer, StorageTek Fellow of Grid Computing, Director of the Grid Computing and Distributed Systems (GRIDS) Laboratory at the University of Melbourne, Australia. He has co-authored the following books: *Microprocessor x86 Programming* and *Mastering C++*; and he has edited *High Performance Cluster Computing*, Prentice Hall, and *High Performance Mass Storage and Parallel I/O*, Wiley/IEEE Press. Dr. Buyya has pioneered Economic Paradigm for Service-Oriented Grid computing and demonstrated its utility through his contribution to conceptualization, design, and development of Service-Oriented Cluster and grid

computing technologies. For further details, visit his Web site at <http://www.buyya.com>.

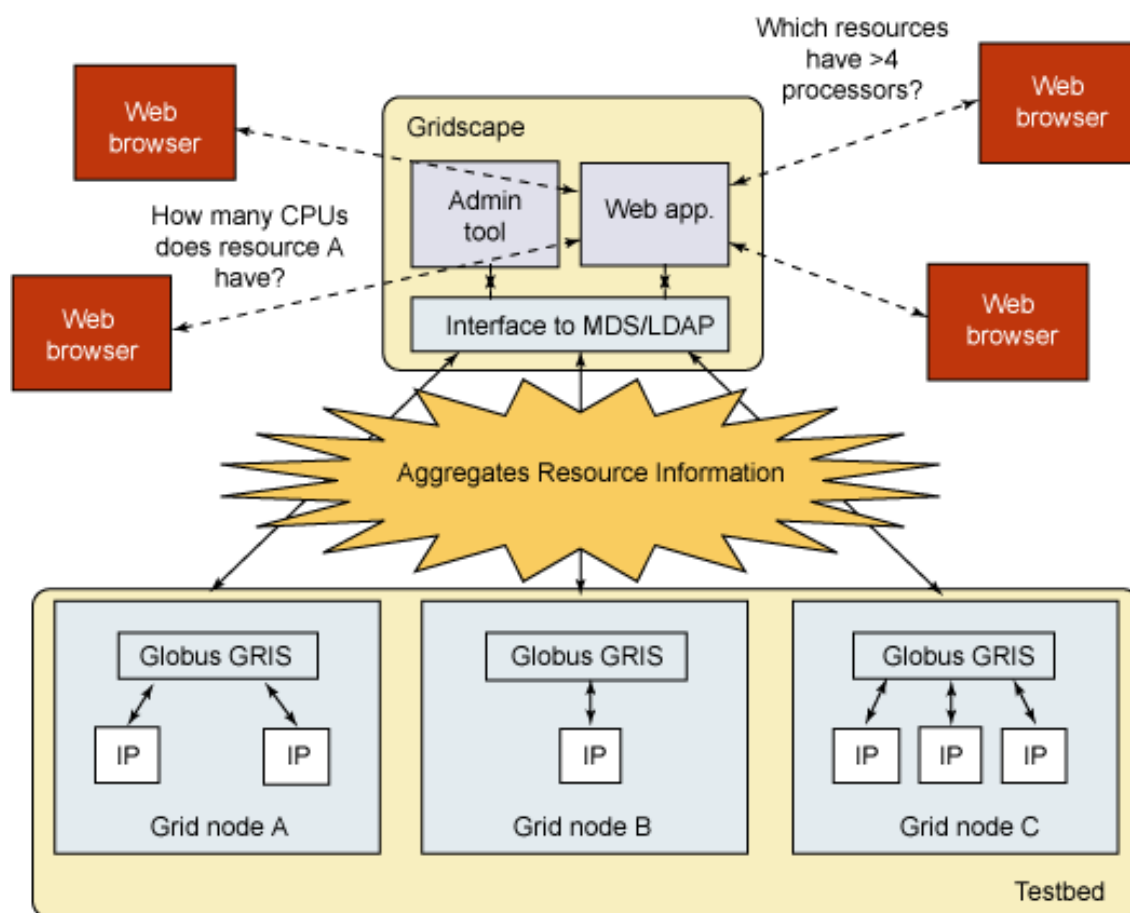
For technical questions or comments about the content of this tutorial, contact the authors or click Feedback at the top of any panel.

Section 2. Introducing Gridscape and Globus MDS

The Gridscape architecture

Gridscape is made up of three components: the Web application, the administration tool, and the interface to the grid information service. An illustration of all the Gridscape components and how they interact with various Grid components is shown in Figure 1.

Figure 1. Gridscape architecture



Web application: The Web application consists of a customizable template portal that provides an interactive graphical view of resource locations and the ability to monitor its status and details. It is also able to submit queries to identify and display only the resources with specific characteristics.

Administration application: The administration tool provides a simple and user-friendly way to create, customize, and update your personal testbed portal. You can manage the resources to be displayed in the portal by adding,

removing, and editing their details.

Interface to the Grid Information Service: Gridscape gets its information about individual grid resources by accessing the Monitoring and Discovery Service (MDS), which is provided by the Information Services component of the Globus Toolkit. MDS is designed to provide a standard mechanism for publishing and discovering resource status and configuration information. It provides a uniform, flexible interface to data collected by lower-level information providers. The "interface to MDS" component was developed as a Java-based class containing high-level methods that hide low-level details (such as LDAP protocols) for accessing MDS services. This level of separation of low-level MDS access mechanisms from other Gridscape components will ensure their portability. For example, if there is a change in MDS access protocols, for example from LDAP-based to XML-based Web services, you can easily update the MDS access interface without having to update other components.

Globus MDS

Gridscape uses the MDS services to gather individual grid node information, so let's take a moment to review briefly the MDS components so you'll better understand how Gridscape interacts with them.

The main function of MDS is to reduce the complexity of accessing system information. To achieve this simplicity, MDS supports local systems that use a wide variety of mechanisms that generate and gather information. However, you need to know only how to interact with MDS to access the information. MDS acts as a point of convergence between the large number of information sources and the large number of applications and high-level services that use them.

Information is represented in accordance to the Lightweight Directory Access Protocol (LDAP) in MDS. LDAP is a set of protocols for accessing information directories. LDAP supports TCP/IP communication and is becoming the standard protocol for dealing with any directory information applications. Using an LDAP server, MDS provides middleware information in a common interface.

Three components make up the MDS hierarchy: Information Providers (IPs), the Grid Resource Information Service (GRIS), and the Grid Index Information Service (GIIS). At the lowest level, the Information Providers (IPs) contain resource data such as current load status, CPU configuration, operating system type and version, basic file system information, memory information, and type of network interconnect. These IPs interface from any data collection service and report to GRIS. The GRIS runs on a resource and contains the set of information relevant to that resource, as provided by the IPs. Individual resources can then be registered to a GIIS, which combines individual GRIS

services to provide an overall view of the grid. You can explore and search the GIIS to find out information about resources.

Interaction between Gridscape and Globus MDS

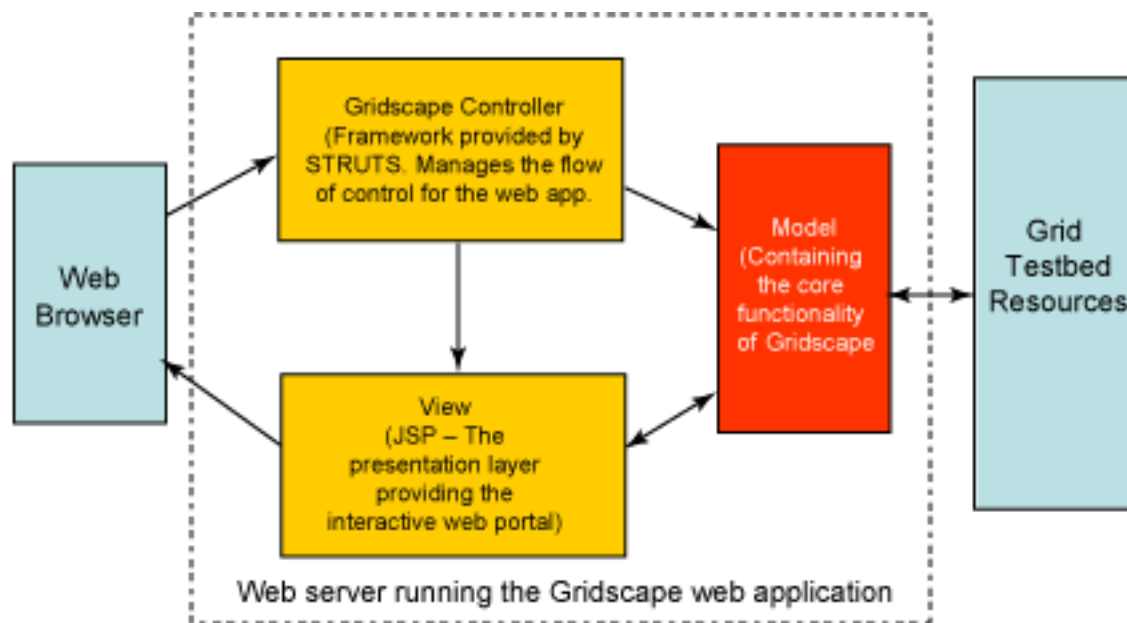
Gridscape obtains the properties of individual resources of a particular testbed through MDS queries to individual GRIS installations. Gridscape also caches the results for later use. It aims to be free of third party tools such as databases. Because continually querying distributed resources is very costly, Gridscape caches the current status of the testbed and allows this store to be shared by any Web browsers accessing the portal. You can periodically update the current status information that Gridscape holds, or you can request an immediate status update at any time.

The fact that Gridscape accesses the GRIS to collect details of individual resources shows that it behaves as a GIIS to a certain extent. It collects separate GRIS information from various resources in order to provide a holistic view of a grid testbed.

Section 3. Design and implementation

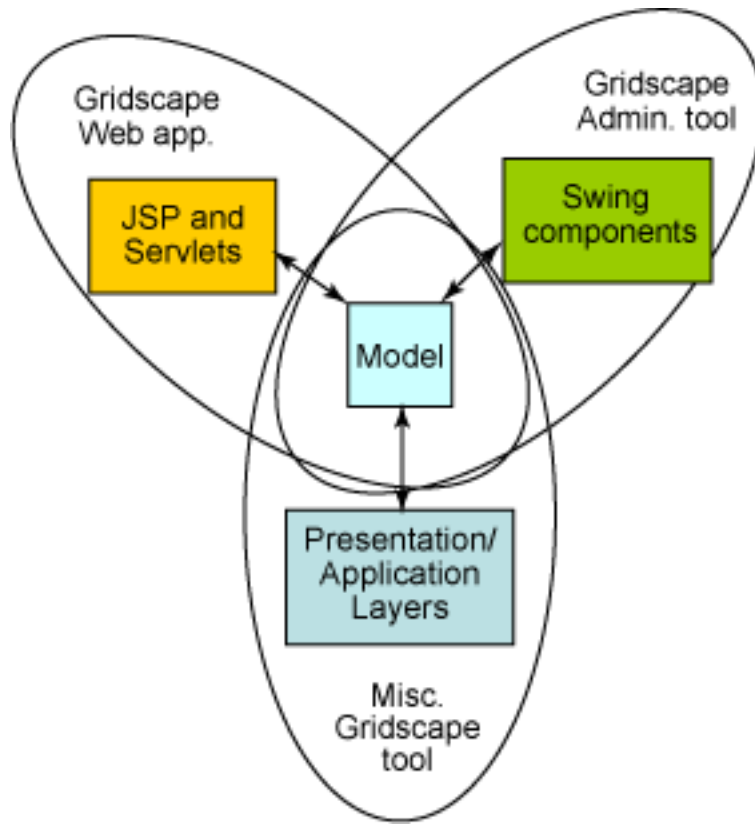
The first component of Gridscape, which is the Web application, follows the Model-2 type of architecture based on MVC (Model-View-Controller). See Figure 2. This architecture provides a means of decoupling the logic and data structures of the application (server-side business logic code) from the presentation components (Web pages). It was also developed for use with Web applications implemented in technologies such as Java Server Pages and Servlets which makes life much easier for developers. In order to make implementation simpler and enhance reliability, Jakarta STRUTS is used to provide a framework for building Model-2 type Web applications.

Figure 2. MVC Model-2 architecture



As shown in Figure 2, the model component contains the core functionality of Gridscape. Because of this separation of the presentation, the control, and the business logic in the application's architecture, we are able to leverage the Model component from the Web application and reuse it in the Gridscape Administration Tool. In this way, programmers can develop other applications that also make use of the core functionality provided in the Model. They can offer a new presentation and control or application layers which access these core components, as illustrated in Figure 3.

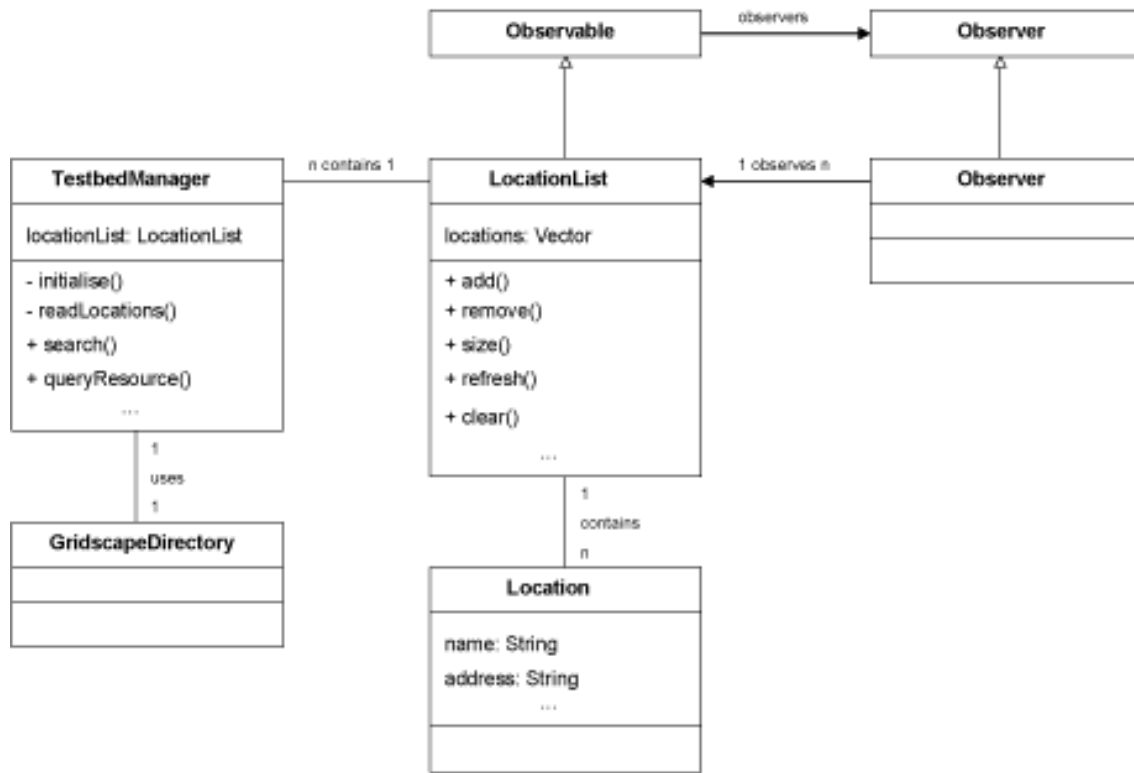
Figure 3. Flexibility and reuse of the Model component



The model

Up to this point, we have identified the significance of the Model and seen its flexibility. However, the Model itself consists of a number of important classes. In this section we will take a closer look at some of these classes, their properties, and their interactions with one another. Figure 4 diagrams these elements.

Figure 4. Class diagram of the core classes of the Gridscape model



GridscapeDirectory

The GridscapeDirectory class provides a convenient wrapper around the necessary elements of the naming and directory access packages of the core Java API. It provides an easy means of connecting to and querying the resources in the testbed. This class will be invoked whenever you try to communicate with the testbed resources, whether it is through the Web application or the admin tool.

Location

This low-level class, Location, is used to represent and hold information about a particular resource. This information includes properties such as the name, address, and port number as well as the MDS data which is gathered from the resource.

LocationList

The LocationList class is used to hold a list of the various locations in a testbed. This class extends the Observable class provided by Java language. When this class becomes observable, the task of developing presentation layers or views that depend on this data is easier.

TestbedManager

The TestbedManager class can be considered the main component because it manages all the other components within the Model and is responsible for handling queries, which are communicated from components outside of the Model. It handles the initialization of the core of the application and handles duties such as searching by allowing other components to collaborate. It is interesting to note that while the TestbedManager contains an instance of the LocationList, it contains only a singleton instance. The benefit of this type of implementation is that through the Web application, even though each client accessing the application is given a new instance of the TestbedManager, there is only one instance of the data. This means that information retrieved from testbed resources is cached, making the site more responsive and ensuring that everyone is seeing the same up-to-date data.

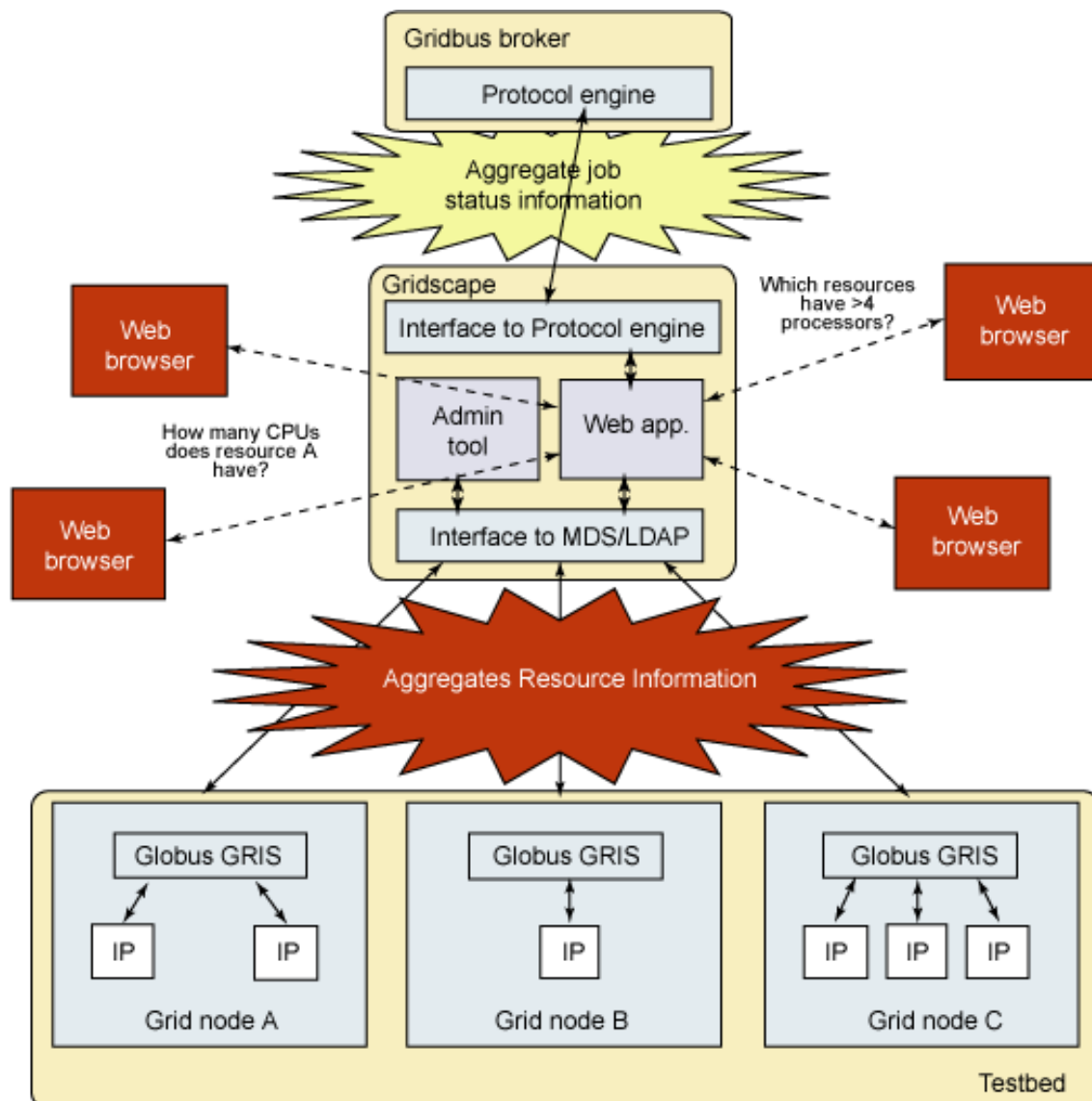
Section 4. Case study

Case study

In this section, we walk you through the steps to create a portal for your own Grid testbed using Gridscape. We illustrate the steps with an example based on the World-Wide-Grid (WWG) testbed used in the HPC Challenge at SC2003. Note that the version of Gridscape used here is slightly different as we have made extensions to increase its functionality. The ability to extend Gridscape is clearly outlined in Section 3.

This version of Gridscape can monitor the status of jobs sent from a specified Gridbus broker to the testbed. The monitoring uses text queries sent to the Protocol Engine which is a component in the Gridbus broker that can provide job location and execution status of each job. Figure 5 is an updated illustration of Figure 1 shown in Section 2.

Figure 5. Extended Gridscape architecture



To begin using Gridscape, you must first deploy the Web application and also install the administering tool.

Deploy the Gridscape Web application

Before you can begin deploying Gridscape, you must make sure that you have Jakarta Tomcat and Jakarta STRUTS installed and working:

- Download the Gridscape Web Archive (gridscape.war).
- Place the Gridscape Web Archive into the Tomcat Webapps directory.
- The application will be deployed automatically when Tomcat is running.

If you have Tomcat up and running, you should be able to view the template portal online.

Install the Gridscape Admin Tool

To easily customize your testbed portal, you should install the Gridscape Admin Tool:

- Download the Gridscape Admin Tool archive.
 - Extract the files into a new folder.
 - Run the admin tool.
-

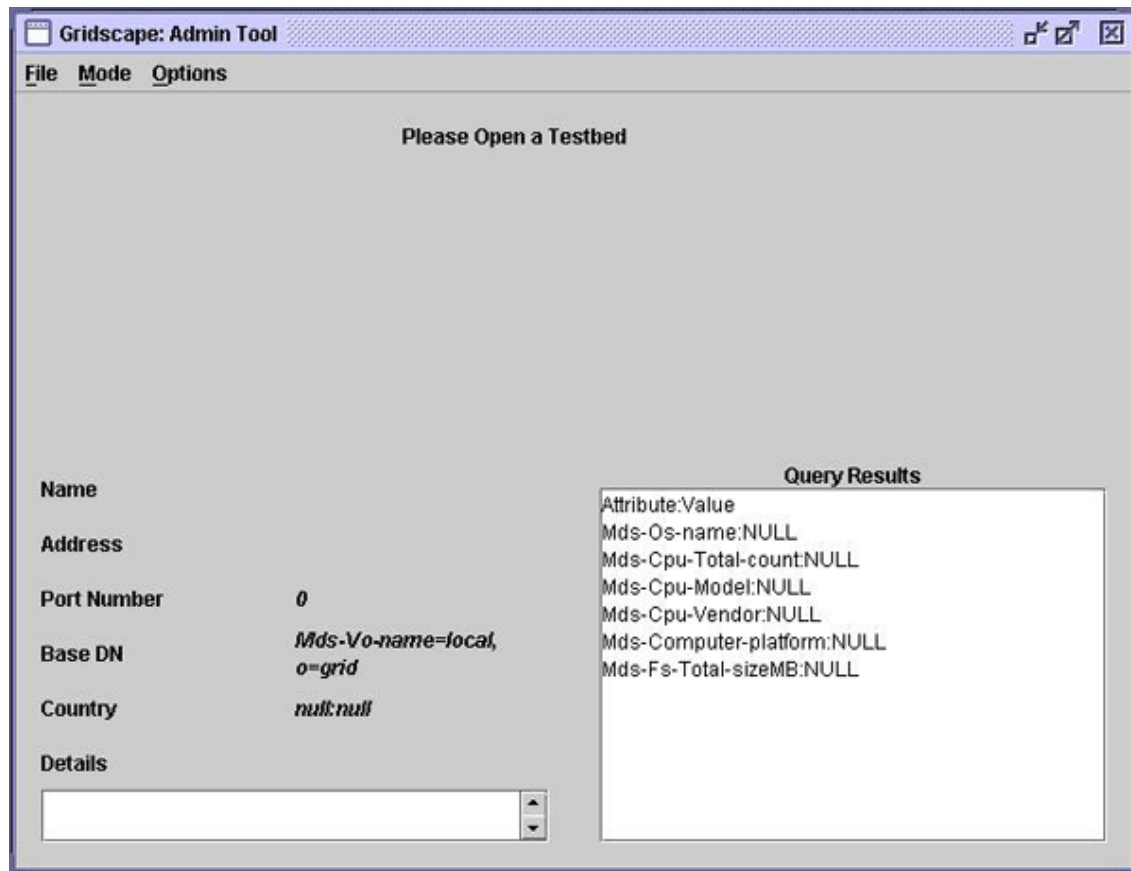
Create your portal

Assuming that you have successfully installed the Gridscape template Web Application and the Admin Tool, you can begin to create your own testbed portal which simply involves customizing the blank template portal that is provided with Gridscape. Most of the details about your testbed are stored in configuration files which can be edited manually. However, in order to make customization easier, Gridscape provides the admin tool.

Launch the Gridscape Admin Tool

The first step to creating your portal is to launch the Gridscape Admin Tool. As this is the first run, you will be asked to open a testbed, as shown in Figure 6.

Figure 6. Initial run of the Gridscape Admin Tool

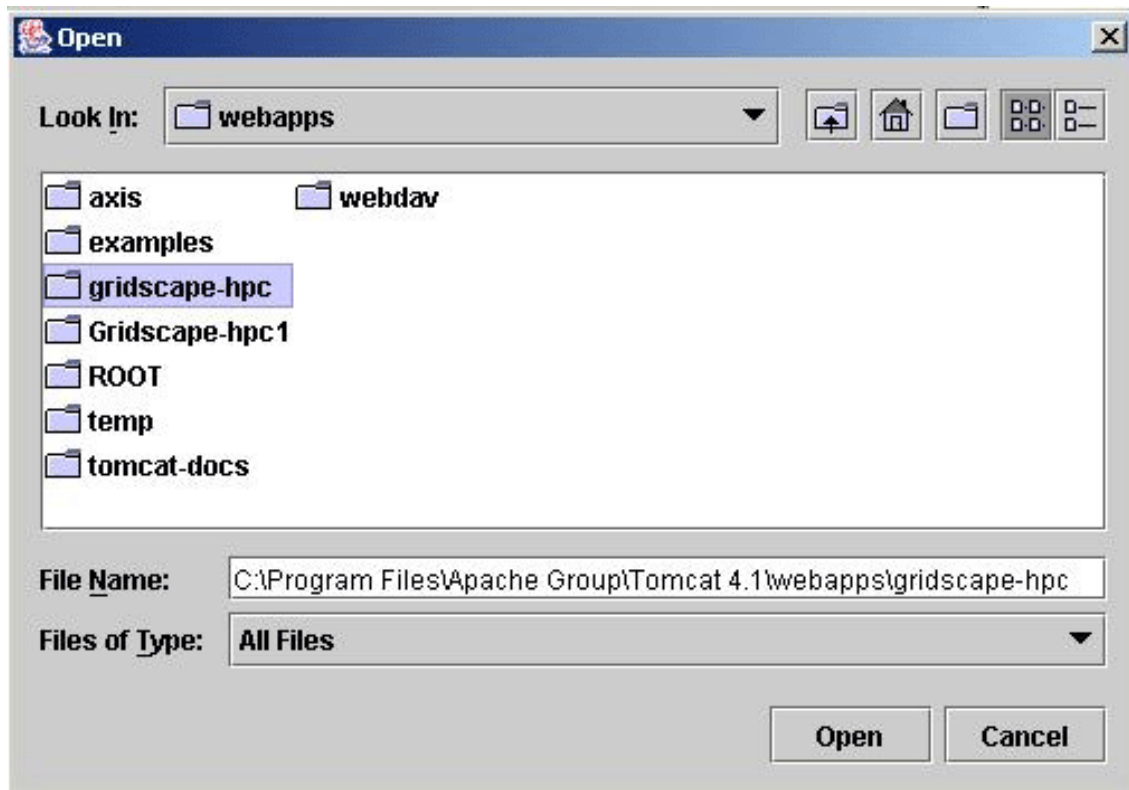


Open a testbed

The next step is to open a testbed. You may already have an existing testbed which you can open for editing, but for your first portal, you should edit the template provided with Gridscape as follows:

- ° From the **File** menu, select the **Open** item. This will bring up a dialog box as shown in Figure 7. You can locate the testbed portal that you want to edit. In this case, it will be the Gridscape template.

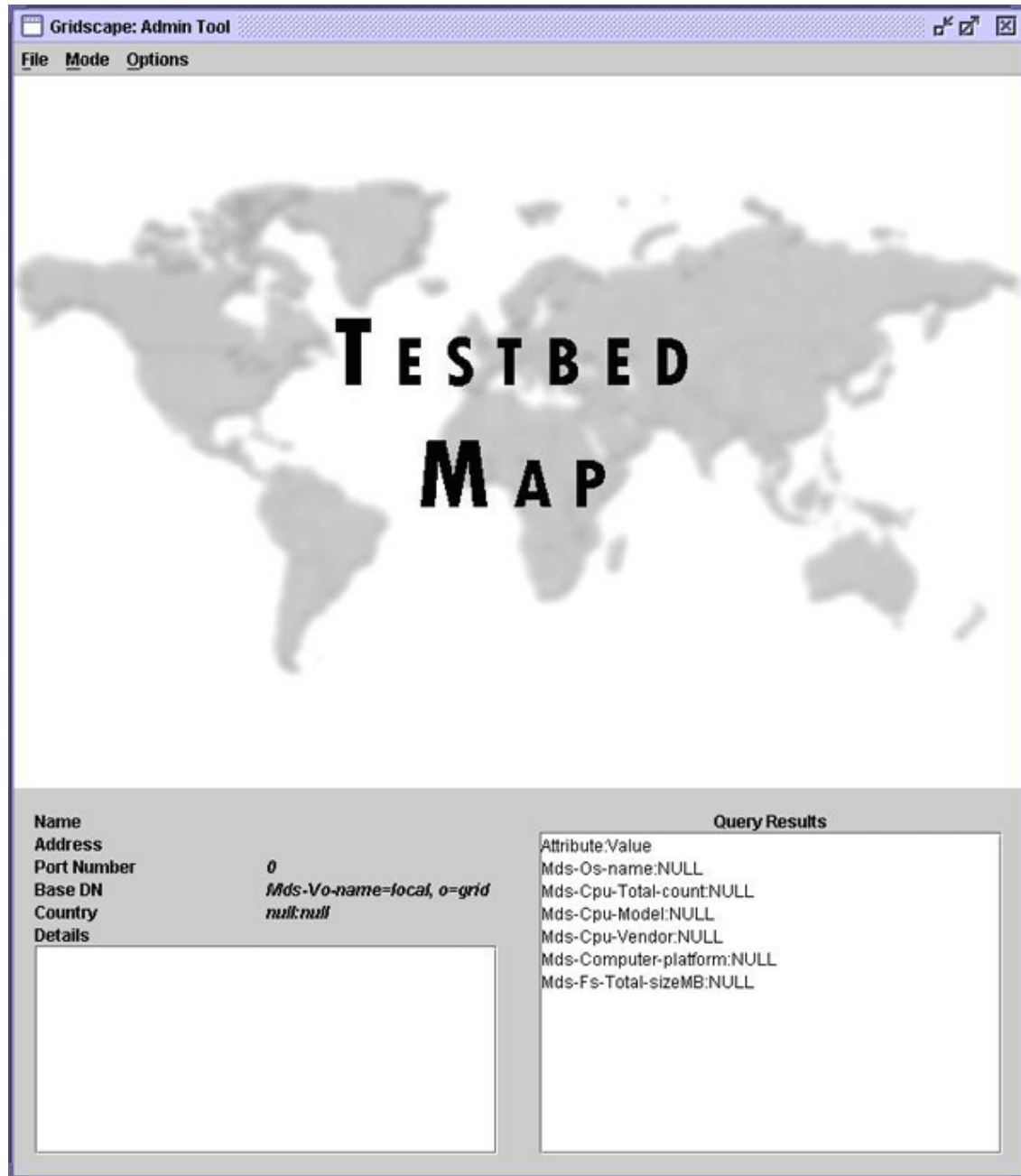
Figure 7. Open dialog



- Locate the Gridscape Web application which was deployed under Tomcat.
- Select **Open**.

The Admin Tool opens the Web application, and the interface changes to display the details of the testbed you have chosen to open. Figure 8 shows the opened template Web application, ready for customization.

Figure 8. Opened testbed template



Changing the name of the testbed

The first step to customizing the template testbed portal is to change its name as follows:

- From the **Options** menu, select the **Testbed Name** item.
- Enter your preferred name for your testbed. Here, we have entered the

name as **World Wide Grid: Global Data Intensive Grid Collaboration Network**, which is the name we selected for the HPC challenge as shown in Figure 9.

Once you have saved the new name, the Web application will be updated immediately.

Figure 9. Testbed name dialog



Change the testbed logo

You should also choose a testbed logo, which is displayed next to the testbed name in your testbed portal.

- From the **Options** menu, select the **Testbed Logo** item.
- Select the **Browse** option.

A dialog box displays which allows you to locate a suitable image and also to specify a URL for that testbed logo.

- Find a suitable image.
- Select the **Set** option. Figure 10 shows the logo selected for our World Wide Grid HPC Challenge testbed portal.
- Select the **Save** option

Once you have saved the new logo, the Web application will be updated immediately.

Figure 10. Testbed logo dialog



Change the testbed map

Next, you will have to select the map to be used for your testbed portal. This is the map on which the locations of the Grid resources will be displayed.

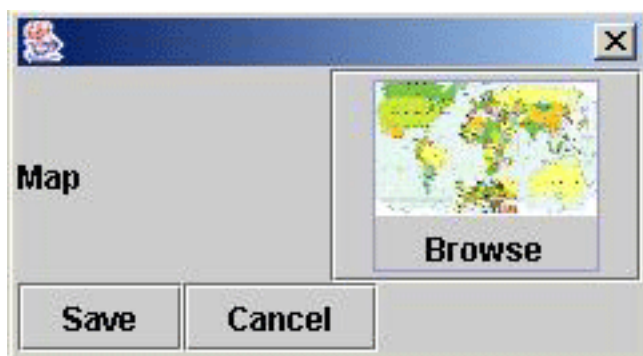
- From the **Options** menu, select the **Testbed Map** item.
- Select the **Browse** option.

A dialog box displays that lets you locate a suitable image to use as a map for your testbed.

- Find a suitable image.
- Select the **Set** option. Figure 11 shows the map selected for our World Wide Grid HPC Challenge testbed portal.
- Select the **Save** option.

Once you have saved the map, the Admin Tool will update to display the new map and the Web application will be updated immediately.

Figure 11. Testbed map dialog



Manage testbed resources

The next step in setting up the testbed portal is to tell Gridscape the details about the resources which will be involved. In Gridscape, we discuss our Grid resources in terms of **locations**, where a location describes not only a single Grid resource, but its physical location as well.

The Gridscape Admin Tool operates in two modes, the View mode and the Edit mode. You can interact with locations in the following ways:

View Mode: The view mode allows you to simply browse and query existing locations. You can select individual locations to query. Once you select a location, its attributes will be displayed in the **Query Results** panel.

- From the **Mode** menu, select the **View** item.

Edit Mode: The edit mode allows you to modify information regarding locations. You can add new locations to your testbed, edit locations that already exist, and remove locations that you have previously added.

- From the **Mode** menu, select the **Edit** item.
-

Add a new location

Now that you have everything set up correctly, the next step to creating your testbed portal is to add your Grid resources (locations).

- Enter **Edit** mode.
- Click on an area of the map where you want to position your location. A new location will be added to the testbed with default values as shown in Figure 12.

Figure 12: Adding a new location

Gridscape: Admin Tool

File Mode Options

Enter name of host [0] Enter Grid Name

Enter Address of host

Enter port number 2135

Select Country --N/A

Select Base DN Mds-Vo-name=local, o=grid Query

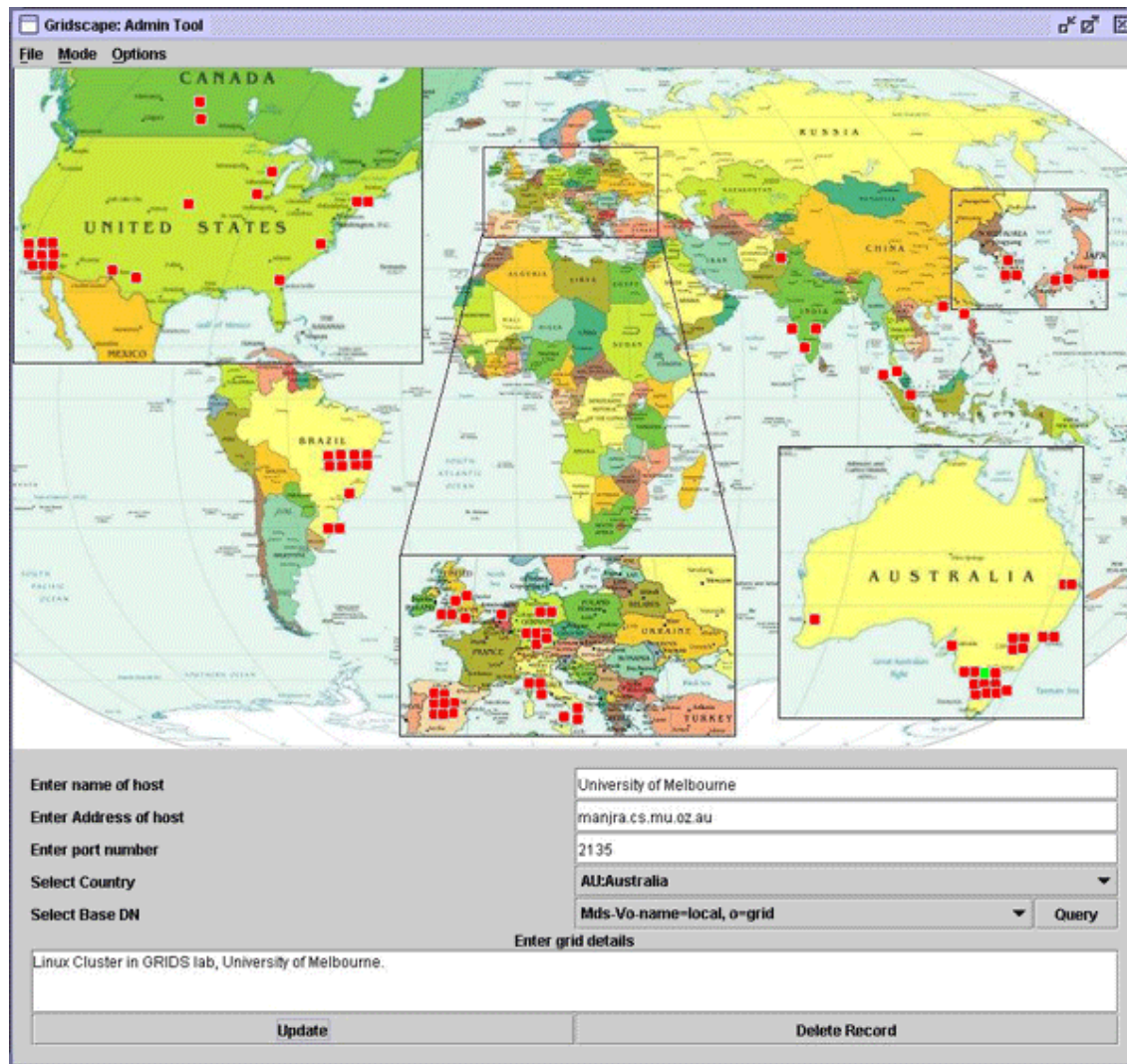
Enter grid details

Description

Update Delete Record

- Enter the name of the location in the space provided.
- Enter the address of the location in the space provided. This address will be used to query the location.
- Enter the port corresponding to the address used to query the location. The default port is 2135.
- Select the country in which this location resides.
- Select an appropriate Base DN (for MDS). The default setting should be appropriate for most instances. However, if this is not the case, you may select **Query** to show a list of available options.
- Enter a short description about the location.
- Choose the **Update** option to store the information just entered about the location.
- You should continue this sequence of steps until you have added all the resources (locations) for your testbed. Figure 13 shows all the locations used for our World Wide Grid HPC Challenge testbed portal.

Figure 13: Updating location information



Move a location

Even after saving, if you notice that one of your locations is not positioned correctly on your map, you can easily move it to the desired place.

- Enter **Edit** mode.
- Find the location that you want to move.
- Click on the location and drag it to the desired position.

You can move your locations around as many times as you want until you are happy with your testbed. Just make sure you remember to save the final

modification.

Edit a location

You may need to edit some of the details of the locations that you have added onto your testbed. Perhaps you made a spelling error or the address of one of the locations has changed. In any case, you will want to edit the details of one or more locations.

- Enter **Edit** mode.
 - Find the location you want to edit.
 - Click on the location.
 - Make the necessary changes to the fields.
 - Select **Update** to store the changes.
-

Delete an unwanted location

If there is a need to remove a location from your testbed, the steps to achieve this are fairly straightforward:

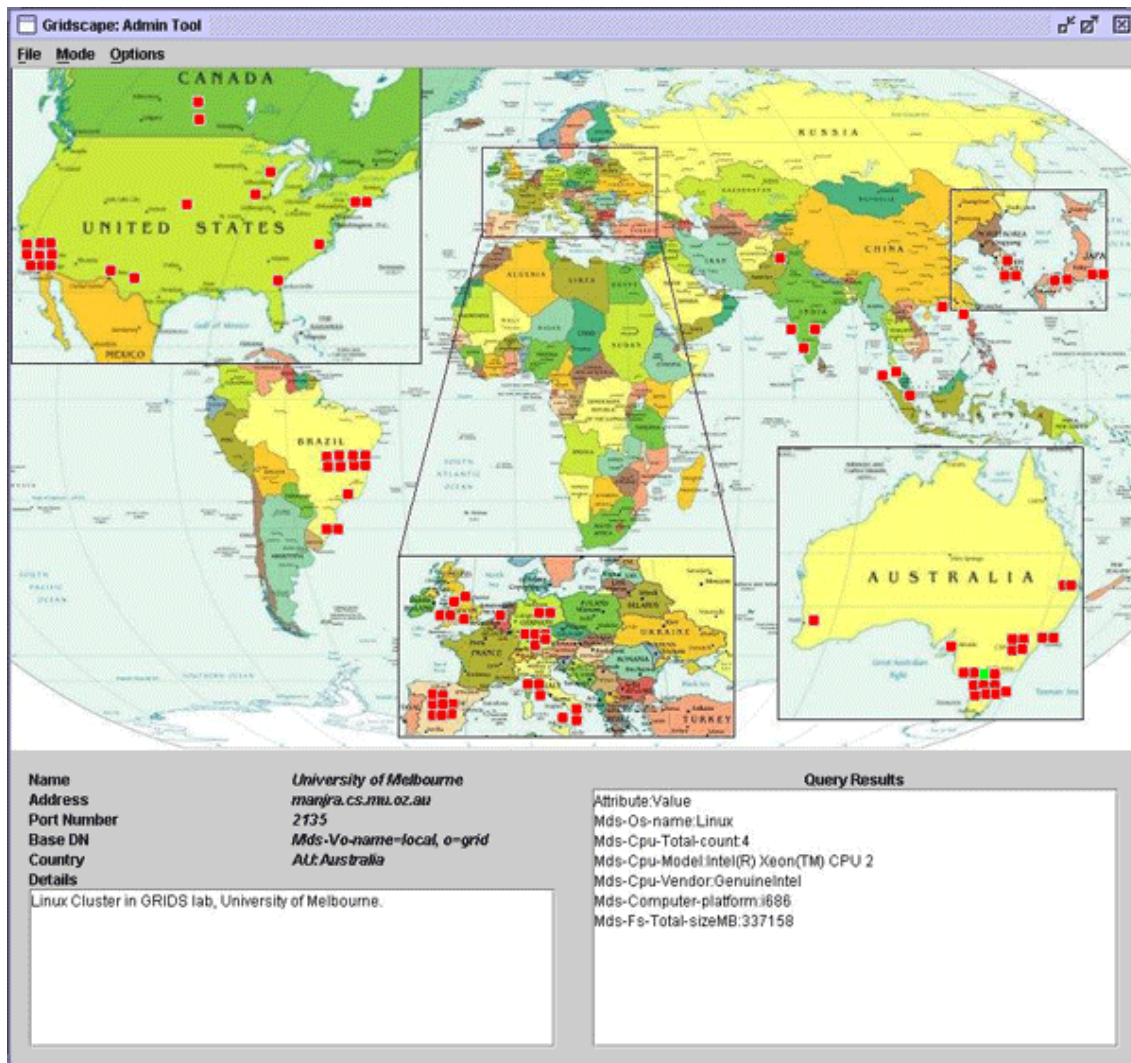
- Enter **Edit** mode.
 - Find the location you want to delete.
 - Click on the location.
 - Select the **Delete Record** option.
-

Query a location

Before saving your changes or viewing the Web portal online, it is a good idea to query the locations that you have just added. This action will give you confidence that the details you entered were correct and will indicate the expected behavior of your Web portal:

- Enter **View** mode.
- Find the location you want to query.
- Click on the location. Figure 14 shows the results of querying one of the Melbourne University GRIDS lab locations in the World Wide Grid HPC Challenge testbed portal.

Figure 14. Querying the GRIDS lab location



Save changes

Once you have added and made the necessary modifications to all the locations for your testbed portal, you will need to save them:

- ° From the **File** menu, select the **Save** item.

Note: If you do not select the **Save** option, your locations will be lost. They will not appear online in your Web application and will not be remembered for the next time you run the Admin Tool.

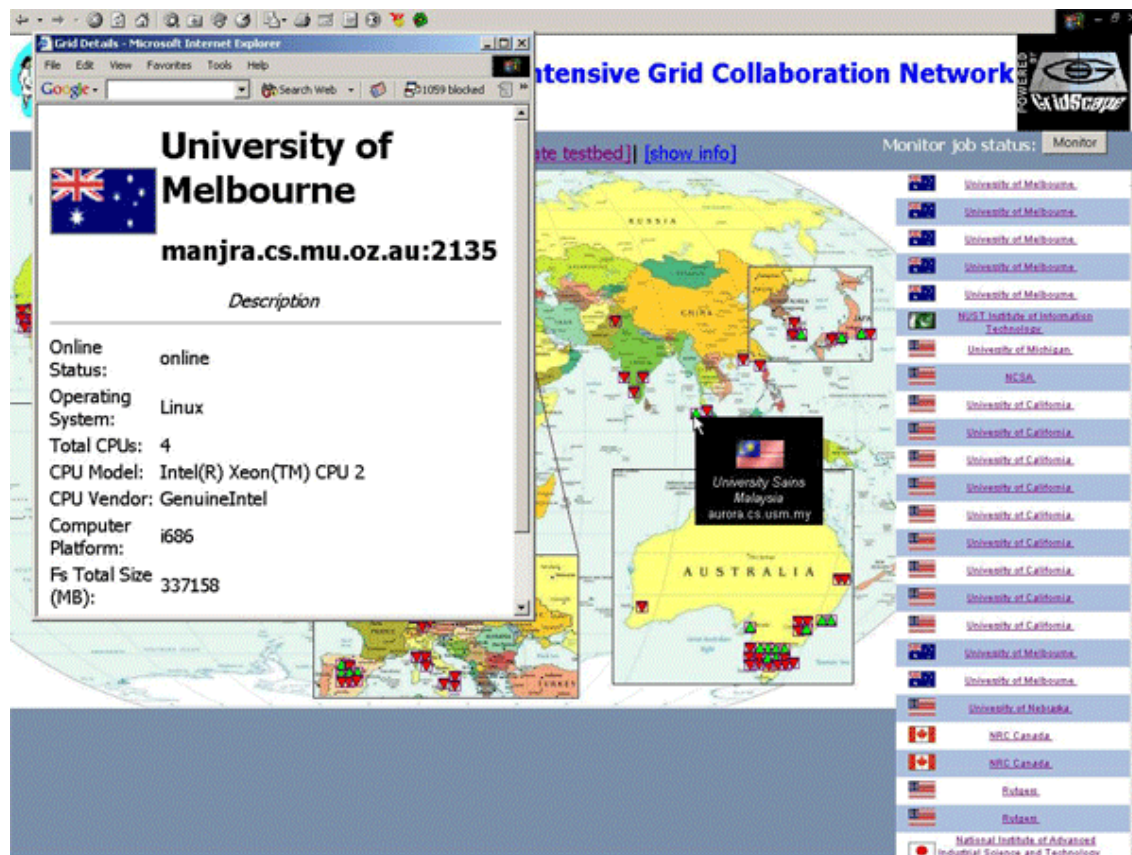
Browse the testbed portal

With the customization complete, the testbed details are saved into a configuration file and deployed on the Web server. You can now browse and monitor the status of the locations on your testbed through a Web browser:

- Start Tomcat.
- Access the portal through the URL:
`http://YOUR_COMP:8080/YOUR_GRIDSCAPE_DIR_NAME/index.jsp`

Hover your cursor over a location to display the location name and country in which it resides. To obtain more specific information, click on the location point itself. A small pop-up window displays with all the details of that particular location. Figure 15 shows a snapshot of the World Wide Grid HPC Challenge testbed.

Figure 15. WWG:Global Data Intensive Grid Collaboration Network



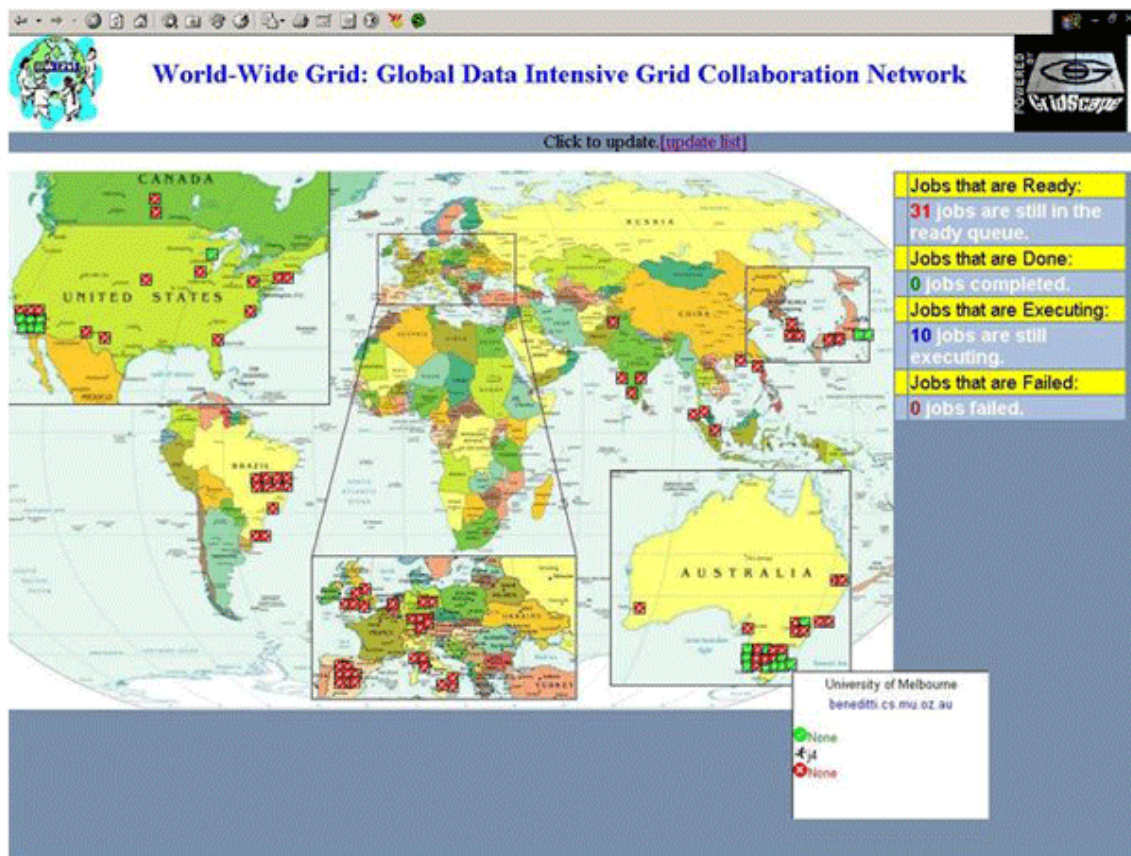
As mentioned early in this section, we have extended Gridscape for the World

Wide Grid HPC Challenge. You can monitor the status of all jobs sent by a specified Gridbus broker to the testbed:

- Select the **Monitor job status** button on the portal.
- Provide the broker URL as well as the port number.

Hover your cursor over a location to display the status of jobs for that particular location. Figure 16 shows a snapshot of this job status monitoring function in Gridscape.

Figure 16. Gridscape extension # job status monitoring function



Section 5. Summary

Tutorial Summary

Currently, a number of unique applications exist, designed to monitor, very specifically, the details of only one grid testbed. This tutorial identifies the need for a tool of this nature # one able to automate the process of creating grid testbed portals. We propose Gridscape, a tool aimed to meet the needs of those who require a testbed portal but simply do not have the resources available to invest in creating their own software from scratch. Gridscape has the potential to provide users with any of the information made available through Globus MDS. It allows for quick and easy creation and administration of Web based grid testbed portals.

Resource List

Several sites will help you learn more about the technologies used in Gridscape:

- You can get [Gridscape](http://www.cs.mu.oz.au/~raj/grids/gridscape/) (<http://www.cs.mu.oz.au/~raj/grids/gridscape/>) at <http://www.cs.mu.oz.au/~raj/grids/gridscape/>.
 - For more information about the [Monitoring and Discovery Service \(MDS\)](http://www.globus.org/mds/mdsusersguide.pdf) (<http://www.globus.org/mds/mdsusersguide.pdf>) , see *MDS 2.2 User's Guide* at <http://www.globus.org/mds/mdsusersguide.pdf>.
 - To find out more about the [components that make up the MDS hierarchy](http://www.globus.org/mds/mdstechnologybrief_draft4.pdf), see http://www.globus.org/mds/mdstechnologybrief_draft4.pdf.
 - The [Globus Toolkit and its components](http://www.globus.org/) (<http://www.globus.org/>) are available at <http://www.globus.org/>.
 - The [Gridbus broker](http://gridbus.org) (<http://gridbus.org>) is available at <http://gridbus.org>.
 - For the [HPC Challenge](http://gridbus.org/sc2003/) (<http://gridbus.org/sc2003/>) # Global Data Intensive Grid Collaboration Demo at SC2003, see <http://gridbus.org/sc2003/>.
-

Feedback

Colophon

This tutorial was written entirely in XML, using the developerWorks Toot-O-Matic tutorial generator. The open source Toot-O-Matic tool is an XSLT stylesheet and several XSLT extension functions that convert an XML file into a number of HTML pages, a zip file, JPEG heading graphics, and two PDF files. Our ability to generate multiple text and binary formats from a single source file illustrates the power and flexibility of XML. (It also saves our production team a great deal of time and effort.)

For more information about the Toot-O-Matic, visit
www-106.ibm.com/developerworks/xml/library/x-toot/ .