A Portlet Service Model for GECEM

Maria Lin and David W. Walker
School of Computer Science, Cardiff University
PO Box 916, Cardiff CF24 3XF

Abstract

Grid-enabled portals are popular as a means to create web-accessible problem-solving environments (PSEs) that allow scientists to access distributed resources, and to monitor and execute distributed Grid applications from a Web browser. In this paper, we describe a web-based portal, based on GridSphere portlets, that deploys services for discovery and management of distributed resources, and for invoking services for mesh generation and computational electromagnetics simulation.

1 Introduction

Computational electromagnetics (CEM) is of increasing importance to the civil and defence sectors. It is central to important problems such as predicting the electromagnetic compatibility between complex electronic systems, and the response of systems to lightning strikes and electromagnetic pulses. These issues are of key concern in possible future platforms such as the More Electric Aircraft (MEA) and the All Electric Ship (AES).

Large-scale CEM simulations are computationally intensive, and can involve access to resources that are intrinsically distributed. For example, in the case of an ”extended enterprise” in which multiple partners from industry and academia are cooperating to design and build a complex system that requires CEM simulations, the geometry of a component may be created at one location, a mesh conforming to this geometry may be generated at a second location, and a CEM simulation based on the mesh may be performed at a third location. Finally, the output from the simulation may be analysed and visualised at one or more other locations.

Grid-enabled computational portals have become a widely-used way of accessing distributed heterogeneous resources and of supporting collaboration between project partners.

In this paper, we present the new design of the Grid-Enabled Computational Electromagnetics (GECEM) portal, extending the work presented at AHM2003 [19], using GridSphere [8] as the portal framework, and describe how it can be used to access heterogeneous distributed resources in the context of the GECEM project.

The GECEM portal has been designed to support the generation of a computational mesh using a meshing service at the University of Wales, Swansea, the solution of a CEM problem on this mesh on a supercomputer at the Singapore Institute of High Performance Computing, and the collaborative visualisation of the results by participants at Cardiff University and BAE SYSTEMS in Bristol [19].

The rest of this paper is organised as follows. Section 2 gives the background to the GECEM project, the motivation of using a portlet approach, and the choice of using GridSphere as the portal framework. Section 3 gives an overview of the design of the

GECEM portal and the components of the portal: the GridSphere Portal, the Grid Portlets web application [7], and the GECEM Portlets for job submission. Section 4 discusses the weaknesses of using the Job Submission Service and the plan to move to use OGSA-compliant services in the next phase of the project. Section 5 compares related portals in relation to the Grid Portlets web application. Finally, Section 6 summaries what has been achieved so far in the GECEM project, and outlines the main areas of future work.

2 Background

The motivation of the GECEM project comes from the fact that resources are intrinsically distributed. In such a situation, the partners in the extended enterprise may be prepared to share data (the geometry, the mesh, the simulation output), but often not the proprietary software systems and the hardware that generate the data (the mesh generator and CEM solver code). Thus, it is not feasible to place all the data, computer, and human resources in one geographical location. The Grid, therefore, is an excellent candidate for providing the infrastructure needed to support extended enterprises. It should be noted that "extended enterprise" has essentially the same meaning as "virtual organisation" [2].

The first version of the GECEM grid was originally implemented using GT2 [19] using a shell script to access the resources remotely, transfer the files, and run the meshes and the solver codes remotely. However, this first version of the GECEM grid has created some difficulties. For example, there is no means of applying for access to all resources. Instead, the user must apply to each site for access.

To make it easier for users to access the Grid resources, in the second version described in this paper, we deploy a grid-enabled portal, specifically a web-based portal environment, for the deployment of Grid services and resources.

2.1 Portlet approach

The new approach presented in this paper involves a number of changes. First, we choose to use a portlet approach. Second, we migrate the shell scripts from GT2 to a Java-based method using the API provided by GridPortlets. Third, we build on the GridSphere portal framework. Fourth, we make use of the portlet service models of GridPortlets to use the Grid resources and to submit jobs. We use the OGSA service-oriented architecture. Fifth, we create GECEM portlets for getting input geometry files, meshes and solvers.

Our choice of using a portlet approach for GECEM project is suitable for a number of reasons. First, the portlet approach is very compatible with the web services model. Second, each service can be associated with a portlet. Therefore, it is very easy to add new services. Third, many different groups can contribute portlets which can be plugged into a portal. Fourth, a user can subscribe and configure portlets according to their need. Fifth, portlets are easy to test and evaluate user interfaces. Sixth, portlets allow customization and separate presentation and logic.

The underlying idea is to use a service-oriented architecture based on emerging standards such as the Open Grid Services Architecture (OGSA). The advantage of using OGSA is that it allows us to provide users with access to GECEM-specific services without creating individual accounts as required by GT2. Thus, the difficulty with the inflexibility of creating individual user accounts in GT2 is reduced as mentioned in [19].

The Open Grid Services Infrastructure (OGSI) [6] specifies how grid services are created and is part of OGSA. The Globus Toolkit 3.0 (GT3), a Java reference implementation of OGSI, provides mandatory Grid service features, such as service invocation, lifetime management, a service data interface, and security interfaces that rely upon underlying Public Key Infrastructure (PKI).

2.2 GridSphere vs. Jetspeed

Examples of portlet based portals includes Jakarta Jetspeed, IBM WebSphere, the Oracle i9AS Portal, BEA WebLogic Portal 7.0 and GridSpere. Among these, only Jetspeed and GridSphere are open-source and free.

The decision to use GridSphere was based on the

evaluation report [11] by the GridLab team, who compared GridSphere with Jetspeed. Based on the lessons and best practices from several notable Grid portal projects, such as the Grid Portal Development Kit (GDPK) and the Astrophysics Scientific Collaboratory (ASC) portal, the GridLab team has developed GridSphere to meet the needs of the Grid community. The design and motivation of the GridSphere project is described in [13]. As pointed out in the evaluation report, Jetspeed has a number of limitations. First, there are too many dependencies on open-source projects such as Turbine, ECS, etc. The developers have to keep abreast not only of the changes in Jetspeed, but also these related open-source projects. Second, codes are not very stable. Third, Jetspeed 1.0 version does not adhere to JSR 168, although Jetspeed 2.0 will. However, Jetspeed 2.0 has not been released yet at the time of implementation.

GridSphere has followed two popular portlet implementations: one is the Java Specification Request (JSR) 168 [17] *de facto* portlet API standard and the other is based upon the IBM WebSphere Portlet API [10]. These two standards are crucial for the stability of portlet developments. JSR 168, being the standard specification for the future, lays a foundation for a new open-standard for Web portal development frameworks. The IBM WebSphere Portlet API is built upon the Servlet API which also allows the concept of Servlet API to be easily used in GridSphere.

An advantage of GridSphere over Jetspeed is that the former is JSR 168 compliant and provides support for localization. It also supports third-party portlet packaging and deployment, which means that it supports the development of re-usable portlets and portlet services. Hence, the portlets can easily invoke the portlet services developed by others. Furthermore, it also provides template project build support for the creation of new portlet web applications, and support for ‘a ‘portlet service” model for the development of services. The ability to use a custom library for creating higher level visual components could save a lot of development effort. Besides, there is no need to integrate with Velocity, Cocoon, WebMacro so that developers do not need to keep breast of the changes of these software and can decide to use the newest XML/XSL technology.

In addition, GridSphere is Java-based, thus, as a computational portal, it provides scientific researchers a single point of entry from which users can easily access Grid Services. Users are provided with a friendly and easy-to-use interface through a web-browser or even a PDA or other mobile devices.

A major difficulty is that Grid software libraries are always changing, and hence there is a need to track these changes. Since GridSphere uses a white-box framework, developers are required to have some knowledge of base framework class, which requires a large effort at the beginning of the development cycle.

3 The GECEM Portal

The GECEM portal is a problem-solving environment (PSE) composed of a collection of portlets and services. The PSE provides the main interface through which services are accessed. The PSE will support the composition of applications from service-based components, the execution and monitoring of such applications on remote resources, and collaborative visualisation, exploration, and analysis of the application results. In addition, the PSE also provides an interface to meshing services and supports the collaborative visualisation of meshes.

The portal is built using the GridSphere portal server [13, 8] and GridPortlets, which were both developed by the GridLab team. The advantage of coupling GridSphere and GridPortlets is that they fully support OGSI Globus Toolkit 3.X. In addition, GridSphere and OGSA can be hosted under the same container to allow a portal to host grid services.

Figure 1 presents the portal architecture in which users logon to GridSphere and configure their MyProxy credentials using GridPortlets. The GECEM portlets are then used to select the resources and files needed to execute a job and then submit it.

There are three types of portlets: GridSphere portlets, GridPortlets, and GECEM portlets. Users of the GECEM portal can use the subscription portlet provided by GridSphere to subscribe or unsubscribe the portlets.

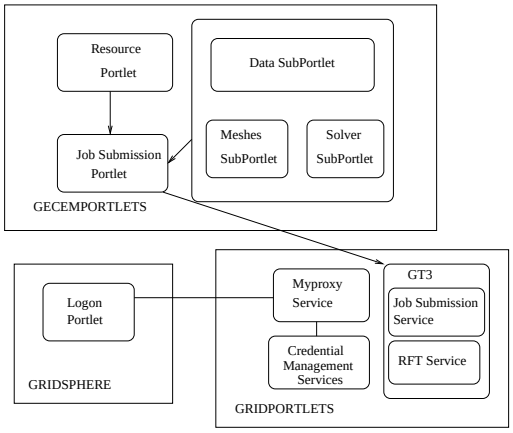

Figure 1. GECEM Portlet Service Model

3.1 GridSphere

The GridSphere portal provides a portlet container, and a set of core portlets and portlet services that provide the basic framework for user management, session management and group management, as well as layout customization and portlet subscription. Specifically, it offers a user login portlet, a subscription portlet, a notepad portlet, a layout manager portlet, and a file manager portlet. This infrastructure supports the development of third-party portlets. It also provides built-in support for role-based access control (RBAC), separating users into guests, users, administrators, and super users. The availability of a high-level portlet model allows developers to build complex portlets using visual beans and the GridSphere User Interface (UI) tag library.

The GridSphere framework is based on the highly popular "portlet model". A portal web page is composed of portlets that provide "mini-applications" each represented as a visual window similar to applications in any other desktop environment that can be minimized or maximized. The Portlet API has just been ratified under the Java Specification Request (JSR) 168 by the Java Community Process (JCP). The Portlet API is very similar to the Java Servlet API but provides methods for the configuration and management of portlets and specifies a portlet packaging model, thereby making it possible for "portlet providers" to separate themselves from the various portlet container implementations including IBM WebSphere, SUN iPlanet and the open-source Jakarta Jetspeed.

3.2 GridPortlets

A GridPortlets web application, composed of a collection of portlets and services, allows end-users to make use of Grid technologies. The GridPortlets project offers portlets for certificate and credential management, which are used to authenticate the user through MyProxy. A user can view their local credential, renew a credential, revoke certificates, generate a certificate request, and store credentials on a MyProxy server. These portlets are useful for securely authenticating users to remote resources, and for users to delegate their certificates to a MyProxy server for later retrieval and login. By configuring GridSphere with GridPortlets, users can logon using their MyProxy password. In addition, GridPortlets also provide Globus Toolkit 3 services such as Reliable File Transfer (RFT) services [3] to transfer data remotely, and job submission services to execute jobs using grid resources.

GridPortlets will support credential management, job submission, file management and examples of OGSI services as well. In addition, similar to the way that GridSphere offers a portal framework for portlet develpment, the GridPortlets also offer a framework for developing other grid portlets for grid applications.

All Grid access goes through the Java CoG Kit [18] which provides interfaces to Grid functionality including transferring files locally and remotely; executing a remote job on the Grid; authenticating to the Grid, but also provides interfaces to more advanced Grid functionality such as simple job queues. The Java CoG kit also offers protection from many changes in the Grid software.

3.3 The MyProxy Manager

In order for the user to use the grid resources the user needs to contacts the portal server with a valid "Proxy Certificate". First, the user has to request a certificate and stores a proxy certificate in a secure MyProxy Server with a temporary password. When the user logon to the GECEM portal, the user enters

the password and the portal server contacts the proxy server and loads the proxy. The portal server will hold the proxy for the user for a short period of time in the user's session state.

The portlets in the GridPortlets web application are useful for securely authenticating users to remote resources, for users to delegate their certificates to a MyProxy server for later retrieval and login, as well as to help them to in making decisions for scheduling jobs by allowing them to view pertinent resource information.

By using the MyProxy package, users can use the portal to gain access to remote resources from anywhere without requiring their certificate and private key be located on the same machine/device running a web browser. The MyProxy server is responsible for maintaining a user's delegated credentials and proxies, which can be securely retrieved by a portal for later use.

3.4 GECEM portlets

GECEM portlets provide a portlet to select data files and resources, submit jobs to remote machines and transfer files for mesh generation and CEM simulation. The portlet design for GECEM has three main components: resource discovery, file selection, and job submission. These three components are implemented by three main portlets.

- The *resource portlet* allows a user to select a machine in the GECEM grid. The list of machines are hand-edited beforehand in an XML file which is similar to the one in the Grid Resource Registry provided by the GridPortlets web application. This file defines the available services and the machine names responsible for these services. For example, the machines for MyProxy and for Job Submission.

- The *data portlet* then displays the geometry data files, input files for mesh generation, and input solver files available on the selected machine, and allows the user to select from these to specify a complete job.

- The *job submission portlet* will commit the resources and the files selected to access the meshing service and the solver service. The job submission portlet provides a button to invoke a script which will access the Globus Reliable File Transfer (RFT) Service and Job Submission Service provided by GridSphere. The RFT Service transfers files from one location to another using the gsiftp protocol. Both the mesh generation and CEM simulation codes are submitted using the Job Submission Service.

To run a scenario, the user first selects the resources from the resource portlet, then selects the data files, the meshes,and the solvers from the data selection portlet.

3.5 Job Submission Portlet

The Job Submission Portlet is the most important part of the GECEM portal for running the mesher and solver codes.

The portlet has the following interface:

- view the machines selected in the resource portlet

- view the input data files, the meshers and the solvers selected

- submit jobs through two job types provided by the Grid Portlets web application.

- view the output files using `gnuplot`

The selection of file and meshes and solvers will be appended to the file and directory as parameters. These parameters are saved using Gecem Service. In the Job Submission Portlet, these parameters are retrieved from the Gecem Service.

There are two main services used by the Job Submission Portlet. These are the Reliable Transfer File Service and the Job Submission Service provided by the Grid Portlets web application.

The Reliable Transfer Service (RFT) is an OGSA based service that provides interfaces for controlling and monitoring third party file transfers using GridFTP servers. The client defines the input and the output files and is hosted inside a grid service so it can be managed using the soft state model and queried using the ServiceData interfaces available to

all grid services. It is a more reliable and recoverable version of the GT2 globus-url-copy tool.

For the Job Submission Service, the Grid Portlets web application provides two job types suggested by GridLab [7]: one through the Globus Resource and Allocation Management (GRAM) provided by Globus, and another one through the Grid Resource Management System (GRMS) [9] provided by GridLab. In the GRAM approach, an API is available to submit jobs to Globus gatekeepers. The gatekeeper is responsible for starting up a job manager to manage the execution and monitoring of the job.

In the GRMS approach, GRMS is a web service written in Java and running on a Java based hosting environment. The hosting environment consists of Tomcat, which is the servlet container. In the GRMS approach, a resource broker attempts to submit a job on the best resource available based on criteria supplied by the user. The current release is based on Globus 2.X and uses Globus Core Services deployed on resources.

Whatever the job type of the job, each job is given a job ID, that can be used to query the status of the job. Each job has a job handle, job type, job specification, job status, argument specification, environment specification, and resource specification. The user can input the environment variable using the environment specification such as the home directory of the files. When the job returns, users can query the job from the job status whether the job is successful or not.

The arguments and the environment will be translated into Resource Specification Language (RSL) to be input to the Job Submission Service. Requests to the job manager are formatted using RSL [4], which is essentially a set of name=value pairs that describe features of the grid job. Some typical examples are: your executable name, any arguments to the executable, and the name of the machine (or machines) to be used.

Another way is to define the variables as parameters in an XML file and the file is read to the job submitted as defined in [4].

4 Discussion

The GridPortlets web application provides a easy-to-use interface to access the basic Grid services. However, by migrating the shell scripts from GT2 to a Java basis using the API provided by Grid Portlets is not a trivial task.

The current version described in this paper also requires a lot of work which is not much different from the shell script version. However, we think that a better approach is to separate the functions in the Job Submission Portlet and allow more flexibility in choosing meshes and solvers using more of the portlet facilities.

At the time of writing, the next phase of the implementation is not to use GridPortlets to submit jobs to execute the mesh generater and the CEM solver. However, a major change is in job submission and to write the mesh generater and CEM solver as independent OGSA-compliant grid services, which will be invoked through different sub-portlets. This work is published in AHM2004 [1]. This improves the user interface, is easy of use, and hides the complexity of the grid infrastructure. By invoking these services through portlets, a user can have a number of choices to choose their data, the meshes and the solvers. Although the changes means that we use less of the GridPortlets web application, such as the Job Submission Service, we still use the Credential Management Service, the Reliable File Transfer Service and MyProxy Configuration for single logon provided by the Grid Portlets web applicatcion.

5 Related Work

A number of portals have been developed for accessing Grid tools and services using the portlet approach. For example, the Alliance Portal [12], OGCE [16], DOE Portal [15], and GridPort [14].

In the Alliance Portal and OGCE, the portlets are targeted to the construction of a Grid portal – a web portal by which a user can access Grid tools and services. They have developed the following portlets which have similar functions to the GridPortlets web application: a Proxy Manager portlet, a GridFTP client, a GRAM Job Launcher, and a Grid Job Sub-

mission utility. However, both OGCE and the Alliance Portal are based on Jetspeed and CHEF.

OGCE fosters collaborations and shareable components with portal developers worldwide. The future work of the OGCE will move to be compatible to JSR 168 and GridSphere. The aim of the OGCE consortium is to provide a high quality, standards-compliant portal framework and portlet components for building Grid computing environments.

Both DOE and GridPort version 3 (GP3) are also based on Jetspeed. GP3 provides a number of OGSA-compliant core services for authentication, job submission, file and data management, process/command execution, and account and session management. For job submission utilities, GP3 provides two portlets: one for batch job submission and another for Community Services Framework (CSF) job submission. GP3 is based on emerging Web and IT standards such as web and grid services model, thus allowing for easy integration of external services such as the GT3 and job sequencing using CSF.

6 Conclusion and Future Work

We have presented a GECEM Portal based on a combination of GridSphere, GridPortlets and our GECEM Portlet application. Specifically, we have described how the integration of GridSphere for logon and user session management with the GridPortlets web application for providing credential management and MyProxy configuration can authenticate users of heterogeneous resources via a single logon. Finally, we make use of the Reliable Transfer Service and Job Submission Service to submit jobs remotely. Within this portal framework we aim to support job submission and collaborative visualisation, exploration, and analysis of the CEM simulation results. Currently we are integrating the GECEM portlet with a mesher and CEM simulation solver exposed as OGSA-compliant services. We are going to use a UDDI registry to publish and discover the meshers and the solvers [1]. In future development, we hope to incorporate changes to use Web Services Resource Framework (WSRF) [5].

References

[1] Yu Chen, Jason W. Jones, Maria Lin, and David W. Walker. A web service architecture for GECEM. In *Proc. UK e-Science All Hands Meeting 2004*, August 2004.

[2] I. Foster, C. Kesselman, J. M. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for ddistributed systems integration, June 2002.

[3] Globus. Reliable file transfer service. `http://www-unix.globus.org/toolkit/reliable_transfer.html`.

[4] Globus. Resource specification language. `http://http://www.globus.org/gram/rsl_spec1.html`.

[5] Globus. The WS-Resource Framework. `http://www.globus.org/wsrf/`, January 2004.

[6] GridForge. OGSI final specification v1.0. `https://forge.gridforum.org/projects/osgi-wg/document/Final_OSI_Specification_V1.0/en/1`.

[7] GridLab. Gridlab and application portlets design. `http://www.gridlab.org/WorkPackages/wp-4/Documents/GridLabPortlets.pdf`.

[8] GridLab. The gridsphere portal. `http://www.gridsphere.org/gridsphere/gridsphere`.

[9] GridLab. GRMS administrative guide v 1.9. `http://www.gridlab.org/WorkPackages/wp-9/res/stuff/grms1.9-admin-guide.pdf`.

[10] IBM. WebSphere. `http://www.ibm.com/websphere`.

[11] Ian Kelley, Jason Novotny, Michael Russell, and Oliver Wehrens. Jetspeed evaluation. `http://www.gridsphere.org/`

gridsphere/docs/jetspeed-eval.pdf, June 2002.

[12] NCSA. Alliance portal. http://www.extreme.indiana.edu/xportlets/project/index.shtml.

[13] Jason Novotny, Michael Russell, and Oliver Wehrens. GridSphere: A portal framework for building collaborations. In *1st International Workshop on Middleware for Grid Computing*, June 2003.

[14] NPACI. GridPort toolkit v.3. http://gridport.net/index.cgi.

[15] U.S. Department of Energy. DOE portal. http://www.doeportals.org/.

[16] OGCE. Open grid computing environments. http://www.collab-ogce.org/nmi/index.jsp.

[17] Java Community Process. JSR 168 portlet specification. http://www.jcp.org/jsr/detail/168.jsp.

[18] Gregor von Laszewski, Ian Foster, Jarek Gawor, and Peter Lane. A java commodity grid kit. *Concurrency and Computation: Practice and Experience*, 13(8–9):643–662, 2001.

[19] David W. Walker, Jonathan P. Giddy, Nigel P. Weatherill, Jason W. Jones, Alan Gould, David Rowse, and Michael Turner. GECEM: Grid-Enabled Computational Electromagnetics. In *Proc. UK e-Science All Hands Meeting 2003*, pages 436–443, September 2003.