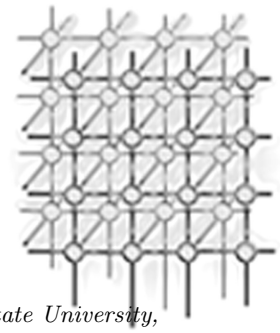
CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2000; **00**:1–7 *Prepared using* cpeauth.cls *[Version: 2002/09/19 v2.02]*

Grid Portal Solutions: *A Comparison of GridPortlets and OGCE*



Chongjie Zhang[1,2,*], Ian Kelley[1], Gabrielle Allen[1,2]

[1] *Center for Computation & Technology, 302 Johnston Hall, Louisiana State University, Baton Rouge, LA 70803, USA.,*
[2] *Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803, USA,*

SUMMARY

This paper will discuss two of the major Grid portal solutions, the Open Grid Computing Environments Collaboratory (OGCE) and GridPortlets, both of which provide basic tools that portal developers can use to interact with Grid middleware when designing their own custom or application specific Grid portals. We investigate and compare what each of these packages provides, discuss their advantages and disadvantages, and identify missing features vital for Grid portal development. The main purpose of this paper is to identify what current toolkits provide, reveal some of their limitations, and provide motivation for the evolution of Grid portal solutions. Application groups should find this paper useful in helping to choose an appropriate Grid portal toolkit for building their Grid portals rapidly in a flexible and modular way.

KEY WORDS: Grid; Portal; Comparison; GridSphere; GridPortlets; OGCE

1. Introduction

The Grid is a complex distributed system that has, at its roots, many differing software packages that form underlying Grid-middleware and infrastructure. The development and deployment of Grid portals that expose Grid services and functionality in a friendly, easy-to-use, and collaborative interface has become popular in the scientific research community. Grid portal toolkits are increasingly being adopted as a means to speed application development since they can provide much of the high-level functionality that is needed to manage the multi-

*Correspondence to: Center for Computation & Technology, 302 Johnston Hall, Louisiana State University, Baton Rouge, LA 70803, USA.
Contract/grant sponsor: Office of Naval Research, Award; contract/grant number: N00014-04-1-0721
Contract/grant sponsor: National Oceanic and Atmospheric Administration; contract/grant number: NA04NOS4730254

Received 29 November 2005
Revised 15 December 2005

institutional and multi-virtual organization issues that arise when designing and maintaining a production portal.

A defining moment in the evolution of portal development toolkits occurred in October 2003 with the introduction of JSR-168 [1], an industry led effort to specify how web components within Java-based portals, called *portlets*, should interact with their hosting environment, or container. A true JSR-168-compliant portlet that does not have any container-related dependencies will be able to run, without any modification, in any number of portal servers including IBM WebSphere, BEA WebLogic, uPortal and GridSphere [2] . The portlet-based architecture provides a flexible and modular way to reuse web components.

There are three major portlet-based Grid portal solutions: GridPortlets[†] [3], the Open Grid Computing Environments Collaboratory (OGCE) [4], and GridPort [5]. In this paper we mainly discuss GridPortlets and OGCE. These two packages provide many of the tools that portal developers need to interact with Grid middleware when building their Grid portals. We investigate what each of these packages currently provides, discuss their advantages and disadvantages, and identify missing features and core functionality that we foresee as vital for the development of successful Grid portal solutions. The main purpose of this paper is to reveal some of the limitations of current toolkits and help identify areas where different groups can work together to drive the progress of Grid portals. This paper can also be used as a reference for application groups to choose a Grid portal toolkit that fulfills the needs of their application.

2. GridPortlets

GridPortlets is an open-source toolkit that was developed at the Albert Einstein Institute as part of the European GridLab [6] project's GridSphere workpackage. GridPortlets provides a high-level interface for portal developers to access a range of Grid services including resource information, credential management, job submission, and file browsing. Figure 1 shows the general architecture of GridPortlets, where a common and clearly defined API abstracts developers from underlying services and middleware. For example, to launch an executable using GridPortlets, a simple *execute* task is constructed that does not require any details about the underlying implementation, which may be the Globus Resource Allocation Manager (GRAM) [7] via the Java Commodity Grid Toolkit (Java CoG) [8] or an advanced resource brokering system such as the GridLab Resource Management System (GRMS) [9].

In addition to providing a high-level API for Grid operations, GridPortlets contains many reusable User Interface (UI) components that can easily be exploited to develop other portlet-based applications. These UI components allow developers to customize the generic JSP pages used by GridPortlets and incorporate them into their applications. GridPortlets itself reuses many of these components for its various portlets, including the file browsing dialog and resource information interfaces.

[†]Although we have tried to make this paper objective, it should be noted that both Gabrielle Allen and Ian Kelley were involved in the European GridLab project, which produced both GridSphere and GridPortlets.

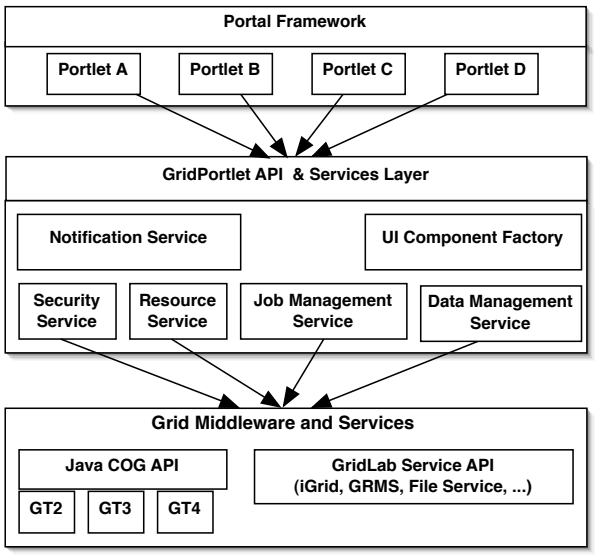


Figure 1. GridPortlet Architecture.

Portal services in GridPortlets are managed through a *Service Registry* that allows developers to "plug in" new implementations to the existing framework. Both services and portlets in GridPortlets can leverage other services to build more sophisticated applications and share data. For example the Job Submission Portlet accesses the MyProxy service to determine if a user has a valid credential, and if not, refers the user to the MyProxy portlet to retrieve a valid proxy.

GridPortlets is packaged with five generic JSR-168 compliant portlets that can be used without any additional development: resource registry, resource browser, credential management, job submission, and file management.

3. OGCE

The OGCE project was established in Fall 2003 with funding from the National Science Foundation Middleware Initiative. OGCE is a open-source collaborative project that leverages Grid portal research and development from the University of Chicago, Argonne National Laboratory, Indiana University, the University of Michigan, the National Center for Supercomputing Applications, and the Texas Advanced Computing Center.

The basis of the OGCE architecture, as shown in Figure 2, is pluggable components in the form of services and portlets. OGCE uses the Java CoG as its main service API for accessing

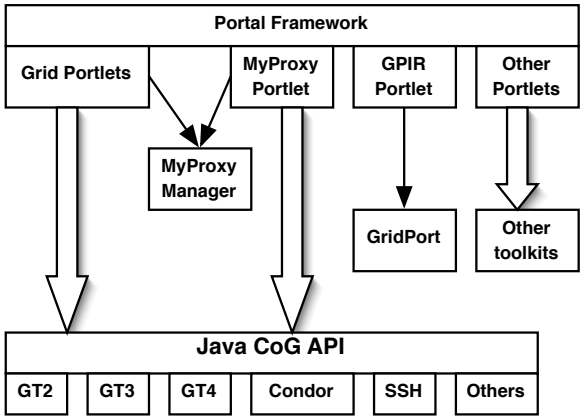


Figure 2. OGCE 2 Architecture

Grid resources. GridPort's Grid Portal Information Repository (GPIR) is used to retrieve local resource information, such as machine loads and queue status.

OGCE comes packaged with services to support Grid operations such as workflow and queue management with the Open GCE Runtime Engine (OGRE) and the Berkeley-Illinois-Maryland Association (BIMA) QueueManager. OGCE provides container-independent mechanisms that allow portlets to share data. For example, the MyProxy Manager allows other portlets accessing Grid resources to use credentials retrieved by the MyProxy portlet.

There are currently two released versions of the OGCE Portal: OGCE 1 and OGCE 2. OGCE 1 used Jetspeed 1.4 as a portlet container and encompassed multiple portal development systems, including CHEF and GridPort. OGCE 1 provided a rich functionality list, including content management, collaboration, and access to Grid resources and information services. However, one of the major disadvantages to OGCE 1 was that its portlets were not compliant to the JSR-168 standard and was therefore not interoperable with other portlet containers.

The OGCE team re-implemented their portlets and delivered OGCE release 2 (OGCE 2), whose portlets are compliant to JSR-168 and can currently be deployed into either GridSphere or uPortal. At the time of this writing, OGCE 2 has not made a final release and is still actively evolving as more portlets from OGCE 1 are migrated into OGCE 2. It provides a core suite of Grid portlets for credentail management, job submission, file management, and GPIR based information services.

4. Comparison

Grid-middleware is constantly evolving as new technologies are developed and more stakeholders become involved in Grid development. It is only natural in this evolution of Grid-middleware that the packages supporting the Grid also will be rapidly changing. Therefore, this paper focuses on the functionality provided by GridPortlets v1.0 and OGCE 2 RC 4. In the remainder of this paper, OGCE refers to OGCE 2, otherwise, OGCE 1 will be explicitly stated.

4.1. Service Layer

Both GridPortlets and OGCE provide information services, job and file management services, and authentication services for portal developers. Information services supply Grid-related resource information, both static and dynamic data. Static data rarely changes and is usually maintained by portal administrators. Examples of static data are machine hostname, machine hardware configuration, machine description, and installed software libraries. Dynamic data, by contrast, is being constantly updated and usually is gathered from underlying Grid resource information technologies such as MDS and includes information such as machine loads and job queues. Unlike information gathering services that are generally automated, job and file services usually require user authentication and control. Both toolkits provide authentication services based on the Globus Grid Security Infrastructure (GSI) [10] and MyProxy.

While overlapping in the basic functionalities of Grid services, GridPortlets and OGCE differ greatly in the service APIs they provide for developers. One key feature of GridPortlets is that it defines a single and consistent high-level API between portlets and underlying Grid services. This uniform API abstracts developers from underlying Grid technologies and infrastructure while supporting multiple implementations. The API can be implemented with Grid programming frameworks, such as Java CoG and Java Grid Application Toolkit (GAT) [11], or with Grid middleware toolkits, such as Globus Toolkit and GridLab middleware services. Developers choose the implementations of particular services during portal configuration or in some cases users are given the choice of which service to use. The GridPortlets API is currently implemented primarily using both Java CoG 1.1 and the Grid middleware services developed in the GridLab project. By contrast, OGCE provides a heterogeneous set of service APIs, aggregated mainly from other projects, such as the Java CoG and GridPort. For job submission and physical file management, Java CoG provides a uniform API that abstracts developers from underlying implementations, including Condor [12], SSH, and different versions of the Globus Toolkit. The details of the service APIs of each toolkit are noted in the following:

Authentication services of both GridPortlets and OGCE use the same underlying technologies: GSI and MyProxy. Both provide similar APIs that allow for retrieval of proxy credentials from MyProxy servers and delegation of credentials to Grid services on behalf of users. The main difference is that GridPortlets stores information on retrieved credentials into a database and allows users to renew them when they expire. OGCE stores credential information in the user's session; once the session ends, all information is lost. Another feature of GridPortlets is that, when used inside the GridSphere portal framework, it provides hooks

to the login portal that allow users to authenticate to the portal using credentials retrieved from a MyProxy server.

Information services of GridPortlets have its own API, containing a resource registry service and information provider service. The resource registry service administers and stores static, although limited, data about Grid resources. The information provider service gathers dynamic data from underlying resource discovery services, currently supporting by default MDS2 [13] and iGrid [14]. OGCE uses GridPort's GPIR as its sole information service. GPIR is a standalone application running with GridPort. It has an administrative client to manage static data and a web-service to gather the dynamic data that is exposed to the portal. GPIR currently supports MDS2 for dynamic data. One noted feature of GPIR is that it persists both static and dynamic data into a PostgreSQL database and allows users to produce historical views on data such as the load of a system over a day, a week, or a month. The GPIR Java client is a powerful tool for updating static information, however, it is currently implemented as a standalone web application that must be installed and used independently from the portal.

Job management includes job submission, job tracking, job migration, and workflow operations. Job submission refers to submitting simple jobs to either GRAM or some resource brokers, while workflow allows users to submit complex tasks that depend on each other. GridPortlets' API supports job submission with GRAM via Java CoG 1.1 and job submission and migration with GRMS v1.9. GridPortlets stores job information into a back-end database. OGCE does not provide its own API for job management, and uses directly Java CoG. In addition, Java CoG can also support workflow via the GridAnt/Karajan Workflow Engine [15], which is directly available to OGCE.

File services have two types: physical and logical. Physical file services know exact file storage locations to execute file-related operations. Logical file services allow users to access storage resources, data, and meta-data transparently without having to know the physical locations of the stored data. For physical file services, both GridPortlets and OGCE use GridFTP and provide similar functionalities and service APIs for file listing and transfer. The main difference is that GridPortlets implements its own API with the Java CoG, whereas OGCE directly uses the Java CoG. Regarding logical file services, GridPortlets' API is currently implemented using the GridLab Data Management Service [16], while OGCE provides services to access the NEES [17] central repository. In the NEES central repository, data is stored on a hierarchal file system with the corresponding metadata and OGCE supports uploading, downloading, ingesting, viewing and querying metadata on the NEES repository.

4.2. Presentation Layer

GridPortlets and OGCE both use JavaServer Pages (JSP) for generating web presentation. Additionally, GridPortlets UI component model provides a number of reusable JSP components, including file browsing and job submission components. These reusable UI components facilitate developers in building interactive, friendly web interfaces. Although OGCE does not supply reusable JSP UI components, it provides tools to support for Velocity-based portlets, which may help developers to port Jetspeed-based portlets into JSR-168 containers.

4.3. Core Grid Portlets

GridPortlets and OGCE both offer Grid-related portlets that can be integrated into Grid portals without additional programming. The major functional features of these Grid-related portlets are compared in the following:

Credential Management - Both GridPortlets and OGCE provide fairly complete operations for managing credentials, including retrieving, viewing, listing, and deleting. However, both packages have some minor weaknesses. GridPortlets does not allow users to freely specify a life-time for a credential retrieved from MyProxy server, which may create problems when a job's execution time is longer than the credential life-time. In OGCE, retrieved credentials are available only within the current HTTP session, requiring users to re-retrieve credentials every time they logon to the portal. OGCE also requires MyProxy servers to allow anonymous credential retrieval, which may pose security problems.

Single Sign-On - This feature allows users to access Grid resources with a single sign-on from portals. To achieve this goal, GridPortlets automatically retrieves proxy credentials for users if they provide their MyProxy password when logging on to the portal.

Resource Information Provider - Both provide rich information about Grid resources, including hardware configuration and job queues. One advantage of OGCE is that its resource browser portlet provides machine load information. GridPortlets provides a simple portlet for managing machine resources by directly editing a XML file. OGCE has a more advanced tool for GPIR administration, however, the client is an external application which must be installed and used independently from the portal.

Job Submission - GridPortlets provides a powerful and friendly interface for job submission that hides users from low-level Grid details and currently supports GT2, GT3 and GRMS v1.9. Compared with GridPortlets, OGCE provides only very basic functionality for job submission with a relatively simple interface. One advantage of OGCE is that users can specify on the portal to use different versions of the Globus Toolkit for job submission, whereas in GridPortlets, the decision of which version of the Globus Toolkit must be made by the administrator during the portal setup. OGCE 1 provided portlets for users to submit jobs to resource brokers, including Condor and the Community Scheduling Framework (CSF) [18], it is expected that these portlets from OGCE 1 will be migrated to OGCE 2.

Job Tracking - GridPortlets stores job history information to a local database and enables users to check a job's status at a later time. Users also can reuse a job specification to submit new jobs. By contrast, OGCE is relatively weak for tracking jobs, allowing users to check only the status of jobs submitted within the current HTTP session. However, OGCE provides the added capability of a BIMA QueueManager portlet to track the status of jobs submitted to queues registered with the QueueManager service.

Workflow Model - Although neither GridPortlets nor OGCE currently support a workflow model, OGCE provides a portlet to view OGRE events. OGCE 1 provided a Java CoG-based workflow portlet for users to setup and submit a workflow and an OGRE portlet to manage complicated Grid tasks through an Apache Ant like task list.

Physical File Management - Both toolkits use GridFTP services and provide rich operations for remotely managing files, including listing directories, copying, moving, uploading, and downloading files. The web interfaces for file management are similar in both projects, the

main difference between them being that GridPortlets provides more operations, including creating directories, viewing text files, and renaming and deleting files.

Logical File & Replica Management - GridPortlets provides similar web interfaces and operations for logical files as for physical file management. In addition to operations such as physical file management, OGCE also provides operations on metadata of data objects, including viewing, querying, adding or modifying metadata.

4.4. Portability and Interoperability

To store job and resource information, Grid- Portlets uses the GridSphere persistence layer, an object/relational API that is implemented with Hibernate. Since GridPorlets' services are based on the service framework of GridSphere, it is not completely independent and, as a result, portlets based on GridPortlets, even though technically compliant with JSR-168, can not easily be deployed into portlet containers other than GridSphere. By contrast, OGCE provides container-independent services and its portlets can be deployed in either GridSphere or uPortal. In addition, OGCE includes a suite of HttpUnit tests for its Grid-related portlets, which can be used to verify the OGCE installation in a portlet container.

5. Additional Features

At the Center for Computation & Technology at Louisiana State University, we are currently involved in the development of two distinct portal projects for coastal modeling [19, 20] and numerical relativity [21]. Through our experience using GridPortlets and evaluating other Grid portal toolkits, we have found that some key features necessary to properly support these user communities are missing. The following list identifies five additions that we would like both OGCE and GridPortlets to consider for future revisions of their software. The list is not exhaustive and highlights a few of the major areas where portal development needs to mature.

Grid Account Management - Provide an integrated system for creating accounts on the portal as well as tying these systems into back-end resources including the integration of management tools for grid-map files.

Certificate Management - Allow users to securely manage their certificate and key files through the portal, including automatically creating user X.509 credentials and delegating them to a credential repository.

Role-based Access Control - Give users and administrators advanced access control mechanisms to manage virtual organizations within the portal. This includes being able to restrict access to resources, applications, and information based upon a user or administrator's roles.

Job Migration and Termination - Use resource brokering systems or other application logic mechanisms to allow users to migrate jobs from one resource to another. Job migration could also help system administrators or software agents to balance machine loads.

Notification Mechanisms - Integration with notification mechanisms within the portal would enable users to subscribe to and be notified when particular events occur (i.e., submitted jobs or file tranfers).

6. Conclusion and Future Work

This paper has discussed two of the prominent Grid portal toolkits currently available. The best choice in a portal toolkit will vary depending on a particular project's requirements, as with choosing any software package.

GridPortlets provides an advanced toolkit for building portal applications, with robust portlets and an extensible services layer. A potential disadvantage of GridPortlets is that it is geared towards base Grid functionality and does not include non-generic or application specific portlets. Adding new functionality to GridPortlets is however relatively straightforward and can be done in a modular fashion.

OGCE packages together an array of portlets and services that can be used immediately as a Grid portal solution. OGCE suffers from lack of extensibility, and many of the components used in its development are not easily reusable or changed. This creates a larger burden on developers who need to add functionality currently not provided by OGCE. In the case where little code development is required, OGCE can prove a good solution for a Grid portal that requires minimal modification.

OGCE and GridSphere are similar in many ways. The portal development community would benefit if these two projects could coordinate development or combine their efforts into a common toolkit that provides the extensibility and easy of development that comes with GridPortlets with the additional collaborative features of OGCE.

Work remains to further develop our comparison, perhaps by evaluating more features (e.g. scalability, stability) or extending the comparison to cover GridPort, and to identify more missing features in current Grid portal solutions.

7. Acknowledgments

The authors wish to thank both the GridSphere and OGCE teams for their comments and discussions. This work was supported by the Center for Computation & Technology at Louisiana State University and the Southern University Research Association (SURA).

REFERENCES

1. JSR 168: Portlet Specification v1.0. The Java Community Process. 2003. *www.jcp.org/en/jsr/detail?id=168.*
2. J. Novotny, M. Russell, O. Wehrens. GridSphere: an advanced portal framework. In *Proceedings of the 30th EUROMICRO Conference*, pp.412- 419, 2004
3. M. Russell. GridPortlets overview. February 2005. *http://www.gridsphere.org/gridsphere/html/mardigrasworkshop2005/02_gridportlets.pdf*
4. Open Grid Computing Environments Collaboratory. *http://www.ogce.org/*, cited in May 2005.
5. M. Dahan, M. Thomas, E. Roberts, A. Seth, T. Urban, D. Walling, J.R. Boisseau. Grid Portal Toolkit 3.0 (GridPort). In *Proceedings. 13th IEEE International Symposium on High performance Distributed Computing*, 4-6, pp.272 - 273, June 2004
6. G. Allen et al. Enabling Applications on the Grid: A Gridlab Overview. In *International Journal of High Performance Computing Applications*, volume 17, pp. 449-466, 2003.
7. Globus Grid Resource Allocation and Management (GRAM), The Globus Alliance. *http://www.globus.org/grid_software/computation/gram.php*, cited in May 2005

8. Gregor von Laszewski, Ian Foster, Jarek Gawor, and Peter Lane. A Java Commodity Grid Kit. In *Concurrency and Computation: Practice and Experience*, vol. 13, no. 8-9, pp. 643-662, 2001
9. J Brzezinski, J Nabrzyski, J Puckacki, T Piontek, et al. Technical Specification of the GridLab Resource Management System, July 2002.
10. R. Butler, V. Welch, D. Engert, I. Foster, S. Tuecke, J. Volmer, C. Kesselman. A National Scale Authentication Infrastructure, *Computer*, Vol. 33, Issue 12, pp. 60-66, 2000
11. G. Allen, K. Davis, T. Goodale, A. Hutanu, H. Kaiser, T. Kielmann, A. Merzky, R. Van Nieuwpoort, A. Reinefeld, F. Schintke, T. Schuett, E. Seidel and B. Ullmer, *The Grid Application Toolkit: Toward Generic and Easy Application Programming Interfaces for the Grid*, Proceedings of the IEEE, 93(3), 2005.
12. Douglas Thain, Todd Tannenbaum, Miron Livny. Condor and the Grid. In *Grid Computing*, John Wiley & Sons, May 2003
13. Globus Monitoring and Discovery System (MDS2). The Globus Alliance. *http://www.globus.org/toolkit/mds/*, cited in May 2005.
14. G. Aloisio, M. Cafaro, I. Epicoco, S. Fiore, D. Lezzi, M. Mirto and S. Mocavero. iGrid: a Novel Grid Information Service. To appear in *Proceedings of the First European Grid Conference*, Springer-Verlag, 2005
15. Sriram Krishnan, Patrick Wagstrom and Gregor von Laszewski. GSFL: A Workflow Framework for Grid Services. In *Preprint ANL/MCS-P980-0802*, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, U.S.A., 2002.
16. GridLab Data Management Services. *http://www.gridlab.org/WorkPackages/wp-8/*, cited in May 2005.
17. Central Repository Software Design Specification. Network for Earthquake Engineering Simulation, 2005. *http://it.nees.org/documentation/pdf/TR-2005-006.pdf*
18. C. Smith. Open source metascheduling for virtual organizations with the community scheduler framework (CSF). *Technical whitepaper*, Platform Computing, 2003.
19. SURA Coastal Ocean Observing and Prediction (SCOOP), *http://www1.sura.org/3000/3300_Coastal.html*, cited on May 2005
20. C. Zhang, C. Dekate, G. Allen, I. Kelley, J. MacLaren, An Application Portal for Collaborative Coastal Modeling, to appear in the special issue GCE05 of Concurrency and Computation: Practice and Experience, 2006.
21. I. Kelley, O. Wehrens, M. Russell, J. Novotny. The Cactus Portal. In *proceedings of APAC 05: Advanced Computing, Grid Applications and eResearch*, 2005.