Journal of Physics: Conference Series

Grid portal architectures for scientific applications

To cite this article: M P Thomas *et al* 2005 *J. Phys.: Conf. Ser.* **16** 596

Institute of Physics Publishing Journal of Physics: Conference Series **16** (2005) 596–600
doi:10.1088/1742-6596/16/1/083 SciDAC 2005

Grid portal architectures for scientific applications

M. P. Thomas[1], J. Burruss[2], L. Cinquini[3], G. Fox[4], D. Gannon[5], L. Gilbert[6], G. von Laszewski[7], K. Jackson[8], D. Middleton[3], R. Moore[6], M. Pierce[4], B. Plale[5], A. Rajasekar[6], R. Regno[1], E. Roberts[4], D. Schissel[2], A. Seth[8], and W. Schroeder[6]

[1] Department of Computer Science, San Diego State University, San Diego, CA
[2] General Atomics, San Diego, CA
[3] National Center for Atmospheric Research, Boulder, CO
[4] Community Grids Lab, Indiana University, Bloomington IN
[5] Department of Computer Science, Indiana University, Bloomington, IN
[6] San Diego Supercomputer Center, University of California at San Diego, La Jolla, CA
[7] Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL
[8] Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley CA
[9] Texas Advanced Computing Center, University of Texas, Austin, TX
Email: mthomas@sciences.sdsu.edu

Abstract. Computational scientists often develop large models and codes intended to be used by larger user communities or for repetitive tasks such as parametric studies. Lowering the barrier of entry for access to these codes is often a technical and sociological challenge. Portals help bridge the gap because they are well known interfaces enabling access to a large variety of resources, services, applications, and tools for private, public, and commercial entities, while hiding the complexities of the underlying software systems to the user. This paper presents an overview of the current state-of-the-art in grid portals, based on a component approach that utilizes portlet frameworks and the most recent Grid standards, the Web Services Resource Framework and a summary of current DOE portal efforts.

1. Introduction

As computational science has matured, an increasing number of scientific communities rely upon third-party software (including both open source community science projects and commercial applications) rather than developing codes from scratch. Lowering the barrier of entry for access to HPC codes and data collections is a technical and sociological challenge: resources and services are complex and distributed; high performance computing (HPC) environments change frequently; and more importantly, users often lack the specialized expertise to deal with the complex HPC environments. Web browser-based portal user interfaces provide access to a large variety of resources, services, applications, and tools for private, public, and commercial entities. Computational science portals have proven to be very useful for many large-scale application science projects. Additionally, Grid-enabled portals [1] can deliver complex grid solutions to users wherever they have access to a web browser running on the Internet without the need to download or install specialized software or worry about setting up networks, firewalls, and port policies. Hence, Grid-enabled portals have been proven to be effective mechanisms for exposing computing resources and distributed systems to general user communities without forcing them to deal with the complexities of the underlying systems. Science portals now exist for a broad range of science collaborations, including efforts in chemistry (GridChem [2], Lattice QCD [3]), astronomy (NVO [4]), physics (Fusion Grid [5], PPDG [6], Cactus [7]), medical/biology (BIRN) [8], nanotechnology (nanoHUB [9]), geophysics (GEON [10], NASA QuakeSim [11]), climate and weather (Earth Sciences Grid/ESG [12], LEAD [13])

This paper presents an overview of the current state-of-the-art in Grid portals with an emphasis on portlet frameworks, a summary of current DOE funded portal research projects and introduces the DOE Portals Consortium, a collaborative effort focused on integrating technologies developed by DOE portal and science application research projects and producing software solutions that can be used across the DOE environment is also discussed.

2. Grid Portal Architectures

Computation and Data Grid portals are often casually confused with project web sites and other forms of Web presence. In this section we review the characteristics of modern portal systems, which typically consist of extensive middleware in addition to user interface development.

2.1. Grid Portal User Interfaces

The user authentication process is one of the distinguishing characteristics of portal systems. In addition to acquiring security credentials for Grid tasks, the login process provides the user with a customized view of the Grid. When a user logs into a Grid portal, it should bring up his or her current "Grid Context," which includes the user's persistent directory of objects, annotations, references and notification logs. It should also provide the set of tools he or she uses to access remote services, configured into groups (typically presented as tabbed browser panes) and organized the way he or she last left them. Access to the user's Virtual Organization group identity within a collaboration gives instant access to group news and communications.

Grid-enabled portals have requirements similar to consumer-oriented portals (Yahoo, CNN, IBM intranet). Services typically include support for a context (login, customization, personalization, user persistent state); support for browser-based user interfaces (they can run anywhere - unlike user desktop clients, connections go through a portal server, so they can overcome firewall/NAT issues); they have "live" dynamic pages available to anonymous or authenticated, authorized users.

Science portals must also support issues related to the integration of Grid-based domain specific applications into the user's environment. Hence, they must manage computations that run for days or weeks on thousands of nodes, and terabyte of scientific data. Specifically, these portals are required to manage credentials, launch jobs, manage files, and hide Grid complexities like batch job submission with RSL across distributed, heterogeneous grids composed of state-of-the-art HPC systems.

2.2. Service Architectures for Grid Portals

Portal services typically include: *Security* - users log onto a portal using a web browser and authenticate by means of a user-id and password (can maintain user, session information). Grid portals map this user ID to grid credentials. *Data Management* – provides access to files, collections, and metadata for local and remote files, supports third party file transfer; *Job submission* - the ability to submit jobs to the Grid for execution and monitoring is a classic service provided by portals; *Information services* - access to directories and status tools is an essential role of the portal; *Application Interfaces* – enables hiding Grid details behind useful application interfaces; *Collaboration* – portals serve as gateways to virtual organizations (VO) to share resources; *Workflow* – presents the user with tasks and assumes the responsibility of integrating these tasks into sequences; *Visualization* - provides tools that give users access to data, rendering, and visualization resources and can provide some level of data viewing or can be used to launch more advanced tools.

Grid portal services must be placed in the larger framework of Service Oriented Architectures (SOA) that are being used to build globally scalable Grid systems. A SOA supports service integration through published and discoverable interfaces. As they are message-based (e.g. SOAP) rather than tied to an application interface they can be integrated in a variety of frameworks. Under the assumption that the message semantics do not change, this model provides a very flexible programming environment. Service providers and clients are free to choose the implementation and backend logic and services independently of each other. The architecture of the Grid has been redefined in terms of a collection of remote Grid SOA services in order to take advantage of this capability. The portal architecture presented in this paper is based on the concept that the portal server is a container of service clients that are designed according to the portlet component model [14]. A portlet represents a window for grouping informational and input values. It is maintained through a Web Portlet engine, which creates, manages, and destroys all the portlets within the portal. Users have the option to integrate portals in their customized view to create an environment most suitable for the individual's user experience. Common use cases for portlets are to maintain access a database, invoke a web service, and to connect to a grid-enabled service [15, 16]. On the administrative side, it is easy to add new portlets. Thus, many different groups can contribute portlets that can be plugged into a portal. Portlet technology is supported and reused by a number of projects and vendors, such as the OGCE, Jetspeed, uPortal, Sakai/CHEF, GridLab GridSphere, as well as most of the major vendors providing portal services including IBM, SUN, BEA and Oracle.

Through the use of a well-defined component model, it is possible to integrate Grid services using a SOA model and to leverage commodity technologies. The advantage of this architecture is that it provides a flexible component-based approach to building portals while being able to access Grid and Web based services. By supplying the user with a basic pallet of portlets that provide the front-end, interactive part of Grid services and applications, the user/developer can customize and organize the environment. The advantage of this architecture for Grid Portals is that it provides a natural way to incorporate "user-facing" Grid services into the portal environment.

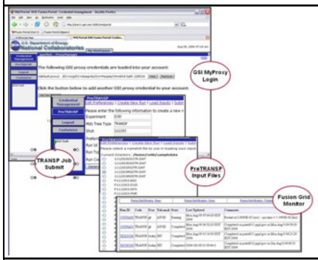
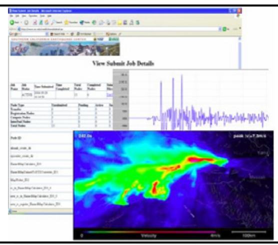
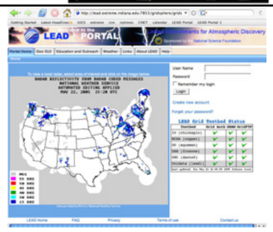
However, using Grid Services is not sufficient to rapidly prototype portals. We still need to write the services themselves and be aware of the potentially changing software standards on the protocol level. Grid abstractions and their implementation (such as those defined by the Commodity Grid Kit (CoG) project [17] help in avoiding this problem. These abstractions allow the application programmer or middleware developer to readily make use of Grid services from a higher-level framework by defining mappings and interfaces between Grid services and particular commodity frameworks. For example, the CoG Kit includes Grid abstractions and workflows that make Grid programming far easier than other Grid middleware and are also a basis for GT4.

Unfortunately, the concept of a single specialized portlet for each remote Grid or web service is not scalable. The user should be able to discover Grid services and dynamically load an interface to interact with it, without having to install a new service client portlet into the server. The new Web Services for Remote Portlets (WSRP) [18] standard is expected to help with this problem. WSRP portlets can be distributed and managed remotely on many servers and the portals composed from WSDL-like information. Clients can select and configure the portlets as needed, and selection can become part of a persistent ``context".

3. DOE Portals Related Research

The portal development activity within the DOE and the SciDAC program for the work described above has been driven by the need of the HPC community to provide scientific gateways to their communities of users, emphasizing the need for a set of tools and software solutions that can be utilized by this community. With this in mind, the SciDAC National Collaboratories (NC) program [19] and the DOE Office of Science has been funding research projects whose common goal is to produce reusable portal toolkits that can be depended to operate within the DOE environment. These efforts are organized as the *DOE Portals Consortium (DOEPC)* that includes the projects *Portal Web Services: Support of DOE SciDAC Collaboratories* and *Middleware Technology to Support Science Portals,* which are developing the core portal services and portlets described in the previous section, *CoG Kits: Enabling Middleware for Designing Science Applications, Web Portals and Problem Solving Environments,* which provides the core Java and Python technology to access the foundational Grid Services, and the *Earth System Grid*, which integrates supercomputers, large-scale data and analysis servers and grids for climate research [12]. DOEPC is closely aligned with the *Open Grid Computing Environments Consortium (OGCE):* the NSF Middleware Initiative (NMI) for Grid Portal Development project, which is a collaborative effort including Argonne, Indiana University, NCSA, San Diego State University, and the Universities of Chicago, Michigan, and Texas.

Table 1. Figures 1-4 describe four portals that either use or significantly influence DOEPC directions.

	**Figure 1.** Earth System Grid (ESG): Primary Goal is to enable community access to climate simulation results distributed across DOE and NSF supercomputers. Features include user registration, authorization groups, file download, browse collection of usage and discovery-level metadata, free-text search, access to data on archival systems, "virtual data services" that provide flexible spatiotemporal subsetting from aggregated collections of files, user management, and metrics reporting. Communities Served: Worldwide climate research community. Over 1200 users registered. Available Data: IPCC, PCM, and CCSM simulations, over 75TB of data and over 400,000 files available. https://www.earthsystemgrid.org/
	**Figure 2.** The Fusion Grid Portal provides a single view of the distributed data collections (MDS-Plus), local user files, and simulation data sets with computational resources. It will be used to run Fusion community codes such as TRANSP and GATO [5]. Based on GridPort/OGCE portlets, the portal allows users to manage accounts and credentials, view status of resources and Fusion Grid jobs, manage files and collections with the File Transfer and Storage Resource Broker (SRB) portlet [20], and submit jobs to the Fusion Grid. The portal uses Fusion Grid CA and MyProxy server.
	**Figure 3.** Southern California Earthquake Center (SCEC) Portal: The SCEC project [21] uses the OGCE portal environment to build an interface to the SCEC digital library that contains seismic wave simulations, observational data, and visualizations. The SCECLib Portal includes portlets specifically developed to display seismic waves. http://webwork.sdsc.edu:10081/sceclib/
	**Figure 4.** The Linked Environments for Atmospheric Discovery (LEAD) Portal: The LEAD portal effort allows users to manage data-driven, high performance weather simulations. These simulations can also be event-triggered, as data mining analysis on the data stream may indicate severe weather events. Detected events trigger simulation workflows on the Grid that are delivered to end users through the Web portal. See http://lead.ou.edu and [13].

In many cases, the software developed by the DOEPC utilizes or has been influenced by other NC projects or Collaboratories including projects that are co-funded by other agencies such as the NSF and NASA. Several of these projects are described in Table 1 above.

4. Summary

The usefulness of grid portal technologies for computational science has been established by the number of portals being developed in the US, Europe, Asia and the Pacific Rim. More importantly, many of these portals are based on frameworks relying on reusable, sharable portlets and add-on technology provided by our consortium. Together these technologies comprise a component architecture that allows the scientific portal developer to build a portal from well-tested and highly configurable pieces. This architecture rests on several major foundational elements: standardized portlet containers based on the JSR-168 standard, Grid standards for security, file transfer and remote execution based on the Globus toolkit and accessed though the CoG programming interfaces and the emerging web-service architecture which allows science applications, hosted on remote resources, to be easily and securely accessed through the portal.

Acknowledgements
This work has been supported in part by the U. S. Department of Energy's Scientific Discovery through Advanced Computing (SciDAC) and the NC projects: Portal Web Services: Support of DOE SciDAC Collaboratories, CoG Kits: Enabling Middleware for Designing Science Applications, Web Portals and Problem Solving Environments; Middleware Technology to Support Science Portals; the Earth System Grid and Fusion Grid Projects.

References

[1] G. Fox, D. Gannon, and M. Thomas, "Editorial: A Summary of Grid Computing Environments." Concurrency and Computation: Practice and Experience, Vol. 14, No. 13-15, pp. 1035-1044 (2002).
[2] The GridChem Project: website accessed on 30-Jun-05 at https://www.gridchem.org/
[3] Lattice QCD Portal: website last accessed on June, 2005 at http://lqcd.jlab.org/
[4] US National Virtual Observatory: website accessed on 30-Jun-05 at http://www.us-vo.org.
[5] Fusion Grid Collaboratory: website accessed on 30-Jun-05 at http://www.fusiongrid.org.
[6] Particle Physics Data Grid: website accessed on 30-Jun-05 at http://www.ppdg.net.
[7] Cactus Project: website accessed on 30-Jun-05 at http://www.cactuscode.org/.
[8] Biomedical Informatics Research Network (BIRN): website accessed on 30-Jun-05 at http://www.nbirn.net.
[9] The nanoHUB Project: website last accessed on 30-Jun-05 at http://www.nanohub.org.
[10] Geosciences Network (GEON): website last accessed on 30-Jun-05 at http://www.geongrid.org.
[11] NASA JPL QuakeSim portal: website accessed on 30-Jun-05 at http://complexity.ucs.indiana.edu:8282
[12] Earth System Grid. accessed On 30-Jun-05 at https://www.earthsystemgrid.org/.
[13] B D Plale, D Gannon, S Graves, D Reed, K Droegemeier, R Wilhelmson, and M Ramamurthy, 2005: Towards dynamically adaptive weather analysis and forecasting in LEAD. Preprints, 2005 Int. Conf. on Comput. Sci ., 22-25 May, Atlanta, GA.
[14] JSR 168 Portlet Specification: website last accessed on 30-Jun-05 at http://jcp.org.
[15] M Thomas, C J Barker, J Boisseau, M Dahan, R Regno, E Roberts, A Seth, T Urban, D Walling. Experiences on Building a Component-Based Grid Portal Toolkit. Accepted for publication in Concurrency & Computation: Practice & Experience, 2005.
[16] D. Gannon, G. Fox, M. Pierce, B. Plale, G. von Laszewski, C. Severance, J. Hardin, J. Alameda, M. Thomas, J. Boisseau, Grid Portals: A Scientist's Access Point for Grid Services, GGF Community Practice document, working draft 1, September 2003.
[17] Gregor von Laszewski, Jarek Gawor, Sriram Krishnan, and Keith Jackson. Grid Computing: Making the Global Infrastructure a Reality, chapter Commodity Grid Kits - Middleware for Building Grid Computing Environments, pages 639–656. Communications Networking and Distributed Systems. Wiley, 2003.
[18] Web Services for Remote Portlets (WSRP): website last accessed on 30-Jun-05 at http://www.oasis-open.org.
[19] DOE National Collaboratories: website accessed on 30-Jun-05 at http://www.doecollaboratory.org/.
[20] San Diego Supercomputer Center Storage Resource Broker: website access June, 2005 at http://www.sdsc.edu/srb/.
[21] Southern California Earthquake Library, SCECLib portal: website last accessed On 30-Jun-05 at http://webwork.sdsc.edu:10081/sceclib.