

Top Ten Questions To Design A Successful Grid Portal

Xiao Dong Wang, Xiaobo Yang, Rob Allan
CCLRC e-Science Centre, Daresbury Laboratory
Warrington, WA4 4AD, UK
{ x.d.wang, x.yang, r.j.allan }@dl.ac.uk

Abstract

With the enhancement of Grid application, Grid portal provides a unique interface for end-users to access distributed resources and are used widely in Knowledge Grid, e-Science, e-learning and e-business. A successful Grid portal relies on many factors. This paper focuses primarily on the design aspects of a successful Grid portal implementation. Ten questions of designing Grid portal are discussed in no particular order. The questions detail the key considering areas of Grid portal architecture, Grid portal contents, portal security, reusability, software design, management and programming test.

1. Introduction

As development of computational science and knowledge mining, the way of using multiple distributed resources to finish a specific work has been used widely by scientists and engineers. A portal is a Web-based application that acts as a gateway between users and a range of such Web services. It provides personalization, single sign-on, aggregation and customization features in addition to other Grid functionality[2][3]. Based on the services provided from distributed resources and end users, portal can be defined as: e-social portal, e-business portal, e-learning portal, Grid portal and Knowledge Grid portal etc. E-social portals focus on providing data and information which social science related, and social science researchers should be the main end-users. The portal powered by NCeSS is a typical e-Social portal (<http://www.ncess.ac.uk/research/small-grants/>). In industry, most companies focus initially on building internal corporate portals that give employees and business users a personalized view of information on the corporate intranet. With the growth of using e-business, e-business portals are developed to enable companies extending portal access to external trading

partners, suppliers and clients which help improve business relationships and communication[4]. E-learning portals provide integrated interface of learning and training application which include Web-based training, virtual classroom and digital collaboration[5]. The e-learning portals are used widely in high education and the students and staff are the end-users. It is not clear to define a Grid portal currently. But basically when a portal is designed for science aim accessing distributed resources and information which are delivered among these resources, we can call it as a Grid portal. The Knowledge Grid portals, extension of Grid portals, are designed on top of the Grid mechanisms and enable particularly on knowledge retrieval, filtering, mining and question-answering techniques[1][12].

The need of more and more Grid portal applications brings a challenge of how to develop a successful Grid portal. It causes interesting of commercial software vendors and open source software developers. On the market, some commercial portals support Grid resource management and application, for example WebSphere Portal (<http://www-128.ibm.com/developerworks/websphere/zones/portal>), WebLogic Portal (<http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/weblogic/portal>), Sun One Portal (http://www.sun.com/software/products/portal_srvr/index.xml), etc... Some open source software developers issue Grid portals to use freely for end-users as well. Among these portals, GridSphere (<http://www.gridisphere.org>) is popular with supporting of Grid Portlets which can access distributed remote resources via Globus Toolkit middleware. Also some activities are organized regularly to provide experience exchange and designing guide for portal developers, for instance a series of workshop on Grid Computing Portals, and workshop of portal and portlets (<http://www.nesc.ac.uk/action/esc/contribution.cfm?Title=261>, <http://www.nesc.ac.uk/esi/events/686>). Even with so many options for the end-users, there are still

possibility to design Grid portals as the applications of different science area have specific features and commands.

This paper will discuss the top ten questions of developing successful Grid portal in no particular order. The questions are hoped to give a guide to the software developers who are keen to design Grid portal and Knowledge Grid portal. The questions detail the key considering areas of Grid portal architecture, Grid portal contents, portal security, reusability, software design, management and programming test.

2. What is the aim of Grid portal?

Many organizations and Grid communities want to develop Grid portals which usually focus on sharing information and centralizing their resources. But before developing Grid portal, they should answer three questions:

2.1 Why end-users need Grid portals

Basically Grid portal acting as information aggregators can access different information resources[7][8]. It is a Web-based application allowing users to access range of different high-level services using a browser interface. Traditionally users can access distributed system resources by logging in the remote servers and processing jobs. This method needs to download and install mechanism to local machine, for example, SSH Security Shell mechanism can be used to implement the login function (<http://www.ssh.com>), also PuTTY is another good selection(<http://www.chiark.greenend.org.uk/~sgtatham/putty>). Using SSH logging in distributed machine and submitting job remotely are suitable for the simple job running in single machine. But if the users' jobs need to be done on several distributed servers or the jobs are more complex to run in parallel, and also if the jobs cause huge data or information transformation among machines, the SSH mechanism is not adequate obviously. A new unique interface should be provided to users. The users can access and manage all of their jobs via the interface without any software download needed. In this case, a Grid portal should emerge and be developed providing to the end-users.

2.2 What aim end-users use portal to do?

Grid portals are used by end-users. It is important to know what they want to get via using the portal and how to provide deep impression for them in Grid portal point of view. Firstly end-users want to understand how

the Grid portal functions. They need to decide if this Grid portal provides suitable functions for their applications. Also they want to feel comfortable when accessing Grid portal, to use and operate the Grid portal simply. And end-users need to be assured their jobs can be run in a reliable, securable and powerful Grid community.

Their main concern is their job running rather than Grid portal itself, so Grid portal should provide easy and simple operation mode for the end-users to implement their complex and complicated job applications. As a powerful and activity Grid portal, more and more information will be supported. It is good, but not all of users need to access all of the information. End-users from different working area focus on different information provided by the Grid portal. So a good designed Grid portal should support personalization. Each user can design his personal page and put useful information in the page. Also Grid portal should allow user to edit and update his personnel page in future.

2.3 Buy commercial portal, use open source portal or D.I.Y.?

Typically the development of Grid portal is based on portal framework. A portal framework provides a possibility for the portal developers to implement Grid application and custom in user's personnel page. Some commercial portal vendors provide high performance portal frameworks, at mean time, some organization release open source portal frameworks for downloading free. Also portal developers can do 'D.I.Y.' to develop their own Grid portal matching their particular requirement. All of these provide more options for the portal developers. Portal developers should realize the difference among the options before deciding their development scenario.

- Basically commercial portals are stable, powerful and provide good after-sale services, but the license fees probably are expensive. Commercial portal can provide enterprise functions, but if the development budget is limited, Grid portal developers may consider to use open source portal framework.
- The open source portal framework is free to download and deploy. As new ongoing specification is usually adopted, the portal framework always releases new versions frequently. It causes the portal developers have to download the portal framework after new version released to benefit the new features support. It even happens more

frequently when the portal framework is in developing. Also some bugs will accompany in the development of portal framework. Portal developers usually have to struggle with the bugs and report bugs to the portal framework organization. It brings extra efforts for portal developers. Another point is portal framework can not provide long term technical support once the framework development work is finished. The typical open source portal frameworks are uPortal(<http://www.uportal.org>), GrisSphere, JetSpeed(<http://portals.apache.org/jetspeed-2>), eXo(<http://www.exoplatform.com>), LifeRay (<http://www.liferay.com>), StringBeans (<http://www.nabh.com/projects/sbportal>), etc....

- If you need a particular Grid Portal to adopt your requirements and functions, and keep usage license, probably D.I.Y. --- developing your own special Grid portal, is another solution. But you should remind that the Grid portal development may spend longer time depending on portal developers abilities and may be expensive than buying commercial portal framework.

3. How to treat the end-users?

The usefulness of Grid portal depends on how many users are using it. We need to recognize that any users using Grid portal can add value to the existences of Grid portal.

- As anonymous or guest users whom visit Grid portal site first time, they are often interested in what the Grid portal can offer but haven't yet decided whether it is worth using or joining in the community via Grid portal. Grid portal should provide a mechanism of showing the functions and features to attract them to use it. That mechanism will not need anonymous or guest users to register before visiting Grid portal and accessing Grid application.
- Grid portal should provide register mechanism for the users whom think your portal is important to them and it is worth to use it. Grid portal also provide registered users by premium contents, the ability to comment on line, interaction with other registered users and portal personalization.
- Furthermore Grid portal should make registered users feel to be part of a community that you want them to be in and encourage

them to attract new users by using the Grid portal. The registration form filled by registered users should be simple and easy, also should allow users to add extra information. After submission, portal sends confirmation back to the users and the registration can be updated by users in future.

4. What services does Grid portal provide?

Grid portals are only as good as the service applications they provide. So if the service applications aren't valuable or aren't interesting to attract users, the design of Grid portals still fail. A good designed Grid portal should consider to put their contents in two categories: core services and additional services.

Core services may consist of execution management, data/file management and information services. Execution management services are concerned with the initiation, monitoring, management, scheduling, and/or coordination of remote Grid computations. The users can put their jobs running on distributed resources. Globus Toolkit supports the Grid Resource Allocation and Management(GRAM) as a basic mechanism for these purposes. Data/file management services are concerned with the location, transfer, and management of distributed data and file. Users can manage their computation results as data and file and put the results in local or remote sites. The typical data/file management services are GridFTP, RFT, RLS, and OGSA-DAI which are integrated in Globus Toolkit, and SRB which is supported by San Diego Supercomputer Center(http://www.sdsc.edu/srb/index.php/Main_Page). The information services can monitor, discover and test distributed resources and services of Grid systems. Users are easily to find out system resources and suitable machines to submit jobs via information services. The typical information services are MDS supported by Globus Toolkit, Ganglia issued by Sourceforge(<http://ganglia.sourceforge.net>) and Inca provided by San Diego Supercomputer Center(<http://inca.sdsc.edu>).

Additional services are not associated with Grid computation normally, but they can add value of your Grid portal. For instance, calendar service may allow users arrange their job schedules properly; e-mail service may notify users' running results via linking with core services; forum provides a platform to ask help from users, collect feedback by portal developers, and announce any road map or release new versions.

5. What kind of portal architecture?

A typical first generation Grid portal architecture consists with three tiers[6]. The first tier is Web browser, Web services and proxy credential services are as middle tier, and the third tier includes backend services and resources, such as databases, high performance computers, storage and specialized devices. But the first generation Grid portal exists some limitations: lack of customization and restricted Grid services. Grid accessing from end-users is statically configured and it is almost impossible to dynamically customize a portal to meet their special needs. As early Grid portals are coupled with specific Grid middleware technologies, it is hard to integrate in other services.

To overcome the above limitations, the second Grid portal architecture is designed as Service-oriented architecture(SOA) and consists of three layers which are portal layer, service layer and resource layer[11]. Portal layer which mainly includes portal framework provides presentation function to the end-users. Service layer implements the calling from portal layer and accesses the physical remote resources. Usually business logic is included in the service layer to invoke different physical resources. Resource layer covers all the distributed resource of the Grid system. It generally includes database, HPC resources, visualization resources etc..

Portlets act as components managed by portlet container to access a set of service resources. Portlet standard provides the possibility of resource reusability[9]. For example, JSR 168 portlet can be plugged and played in Grid portals which support JSR 168 standard without source code modification(http://www.jcp.org/aboutJava/community_process/final/jsr168). The supports of JSR 168 to window status and page mode provide the personalization to the end-users. The Grid portal performance is highly relayed by the implementation of Grid portlets.

6. Is portal secure to use?

Security is an import issue in developing Grid portal. It assures not only no leaking information of registered users and job running results, but also no attacking to the Grid system by network hack. In regarding of the Grid portal architecture, portal layer and service layer usually are hosted in Grid portal server machine, resource layer consists distributed Grid resources which map to remote machines. The communication between service layer and resource layer is standard Web service protocol(SOAP, WSDL) as the SOA architecture design. End-users visit Grid portal via Web browser. So the security should focus

on the communication between end-users and portal layer, service layer and resource layer. Grid portal security should include two parts: data transformation and certification validation. Most data transformation happens between end-users and portal layer, service layer and resource layer. Certification validation includes users' identification and resources usage permission.

Traditional transformation protocol is http which transfers data in plain text type without any security. Https is http in security which adopts Secure Sockets Layer(SSL). SSL is a protocol developed by Netscape for transmitting private documents via the internet. Both Netscape and Internet Explorer support SSL, and many Web sites use the protocol to obtain confidential user information.

Certificates are digital documents attesting to the binding of a public key to an individual or other entity. They allow verification of the claim that a given public key does in fact belong to a given individual. Certificates help prevent someone from using a phoney key to impersonate someone else. Usually a proxy of user's digital certificate is used to validate his identification. But can you create your proxy of certificate easily in any machine which using Web browser to access Grid portal? The answer is no. Thanks NCSA to provide MyProxy as a good solution(<http://grid.ncsa.uiuc.edu/myproxy/>). MyProxy combines an online credential repository with an online certificate authority to allow users to securely obtain credentials when and where needed. It is used widely in Grid community.

7. Can portal tools be reused?

With the need of powerful and complex Grid portal tools, more and more portal developers involve to develop cooperatively. Usually one Grid portal tool development needs several portal developers to write code, test and deploy. So we need a special design model assuring developers working parallel and efficiently. Also the portal tools should be written as reusability for the future development.

Model-View-Controller (MVC) model is proposed by Xerox since 1979(<http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>). The key point of MVC is to separate a Grid tool application's data model, user interface, and control logic into three distinct components, so that modifications to one component can be made with minimal impact to the others. Except JSP can be used to represent View component, JavaServer Faces(JSF) is another new technology to build user interface(<http://java.sun.com/javasee/>)

[javaserverfaces/](#)). It is a standardized specification for building User Interfaces (UI) for Grid portal server-side application. Grid portal developers of various skill levels can quickly build Grid applications by: assembling reusable UI components in a page; connecting these components to a Grid application data source; and wiring client-generated events to server-side event handlers.

As the description in Portal Architecture, JSR 168 Portlet provides the possibility of resource reusability. Web Services for Remote Portlets (WSRP) (<http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf>), is a promising specification that is attempting to standardize presentation-based and interactive portlets. WSRP defines a communication protocol between a Consumer, which is typically a Grid portal, and a Producer, which is a portlet container. WSRP allows functionality that was previously difficult to achieve, such as deploying portlets once, but calling them from anywhere, bringing together third-party portlets, and enhancing interoperability between portals from different vendors. WSRP also gives Grid portal developers a feasible way to build distributed Grid portals. A distributed Grid portal system consists of a network of interoperating Grid portals, whereby services hosted in one Grid portal can be made available to many. This innovation gives WSRP the potential to enhance portal deployment flexibility with access to remote resources.

8. How to communicate and debug source codes?

With the complex Grid application and software development, many developers will involve in and more complicated works need to implement. So we need a method to simplify the complex tasks of software development, transfer structure thoughts, clarify communication, and find right abstraction. Unified Modeling Language(UML) matches above needs(<http://www.omg.org/technology/documents/formal/uml.htm>). UML provides a notation for describing software-intensive systems in an object-oriented way, for instance using a mixture of text and graphics. It is also general enough to describe non-software systems, for example, workflow or other business logic process. Some UML tools can support UML features and deliver the software development forward easily. Among them, Poseidin UML tool(<http://gentleware.com/index.php>) and Visual Paradigm (<http://www.visual-paradigm.com/product/vpuml/>) are quite good UML tools.

Successful projects often are the result of successful organization and communication. During software development and even the software running in Grid portal, there are still some exceptions and errors happened. A mechanism needs to provide for the software developers to produce software and solve any problem, and for users to report any exception during using Grid portal. Bugzilla and Elips are the solutions. Bugzilla is server software designed to help software developers manage software development (<http://www.bugzilla.org>). It allows individual or groups of developers to keep track of outstanding bugs in their produce effectively, is a powerful tool that will help developing team organized and communicate effectively. Elips(<http://opinionplugin.sourceforge.net>) is a light-weight bug tracker which written in Java and JSP.

Another solution to communicate and announce information is to create e-mail group --- mailing list. The e-mail group can be divided in categories. Some categories are opened to public for reporting any problems and inquiring helps. Some categories are private to developers for communicating software development. The mailing list design should allow people to subscribe and unsubscribe by themselves.

9. How to control version software?

Software development needs version control. The history of sources files and documents need to record and retrieve as well. Also developers may work on the same source code together. Concurrent Versions System(CVS)(<http://www.nongnu.org/cvs/>) can allow software developers save and retrieve different development versions of source code. It is a system for managing simultaneous development of files, and is in common use in large programming projects[10]. CVS is designed for developers, either individually or in teams. For individuals, CVS provides a repository from which you can work from home, the office, or the client site without having to haul disks around. For teams, it also keeps a record of who changed which lines of a file and prevents direct overwriting of each other's work. Except CVS, SubVersion is another successful open source version control software(<http://subversion.tigris.org>).

10. How to design code guideline?

With the enhancement of Grid application, more and more software source codes are written as classes and called by other classes. If the software development involves many developers, the developers may write

source codes in same class names but the class functions are totally different. It brings us confused when we call the classes. Furthermore it will cause serious problem if calling the wrong class even the class name is correct.

The solution is to wrap the source code in package with different path. By this method, it is easy to distinguish the class and organize the structure of source code. A package is suggested as “*organisation.country.institution.department.group.portalname.Grid-application-category.portlet-function-name(or utility-name)portlet-name*.”. For example, in NGS Grid portal design, a jobsubmission portlet class path is wrapped in package as *org.uk.cclrc.esc.gtg.ngsportal.jobsubmission.jobsubmissionportlet.class*. it expresses that the jobsubmission portlet is developed by Grid Technology Group(*gtg*) of e-Science Centre(*esc*) in Council for the Central Laboratory of the Research Councils(*cclrc*) at UK. It is used for National Grid Service portal(*ngsportal*). The above description expresses the full information of this Grid application clearly.

11. How to test Grid portal?

Grid portal test is important for designing a successful, powerful and stable project. The test includes software test and operation test. Software test will focus on efficiency and functionality, and it mainly be done by portal developers. Some useful test tips can be used:

- Set break points in conditional path and test if it breaks when the condition happens.
- Run Grid portal in different Web browsers to check if GUI interfaces are compliant with these Web browsers.
- Run as many as Web browsers accessing Grid portal in parallel to test the maximum jobs provided by Grid portal server.
- Stop a job and recover it to test fault tolerance.
- Run Grid jobs from internal and external firewall of Grid portal.

Operation test usually ask early flying birds of users to do. The users can be volunteers from colleagues of developers and from external organization. The users are encouraged to test in several items as below:

- Access Grid portal on the same or different internal or external sites to test scalability.
- Access Grid portal using IP addresses instead of domain names.
- Access Grid portal with huge, complex jobs to test functionality.

12. Conclusions

A successful Grid portal depends on many factors. This paper focuses primarily on the design aspects of a successful portal implementation. Ten questions in designing Grid portal are discussed particularly. The questions cover the key considering areas of Grid portal architecture, Grid portal contents, portal security, reusability, software design, management and programming test. Hopefully it can provide helps and ideas for the portal developers to design a successful, powerful Grid portal.

References

- [1] Zhuge, H.: China's E-Science Knowledge Grid Environment, IEEE Intel. Systems, 19 (1) (2004) 13-17.
- [2] Allan, R., Awre, C., Baker, M., Fish, A.: Portals and Portlets 2003. NESC Workshop, 14-17, July 2003. http://www.nesc.ac.uk/technical_papers/UKeS-2004-020.6.pdf.
- [3] Wang, X. D., Allan, R.: Portlet, WSRP and Application. GridSphere and Portlets Workshop. Edinburgh, 3-4 March 2005.
- [4] White, C.: The Evolution of the E-Business Portal. DM Review Magazine. August 2001 Issue. http://www.mdreview.com/article_sub.cfm?articleId=3822.
- [5] Mulgan, F.: E-Learning = Environment Learning. Learning Circuits. <http://www.learningcircuits.org/2006/April/mulgan.htm>.
- [6] Baker, M., Ong, H., Allan, R., Wang, X. D.: Virtual Research in the UK: Advanced Portal Services. UK e-Science AHM, Nottingham, 2004.
- [7] Thomas, M. P., Burruss, J., Cinquini, L., Fox, G., Gannon, D., Gilbert, L., Laszewski, G. V., Jackson, K., Middleton, D., Moore, R., Pierce, M., Plale, B., Rajasekar, A., Regno, R., Roberts, E., Schissel, D., Seth, A., Schroeder, W.: Grid Portal Architectures for Scientific Applications. Journal of Physics: Conference Series 16 (2005) 596-600.
- [8] Kim, M., Meliksetian, D. S., Malone, C.: A Mainframe Grid Computing Infrastructure. 25 May 2005. WebSphere Journal. <http://websphere.sys-con.com/read/47894.htm>.
- [9] Lin, A. W.: Building a Unified Grid, Part 2: Portlet Interfaces in the Grid User Environment. 29 Nov. 2005. <http://www-128.ibm.com/developerworks/grid/library/gr-unified2>.
- [10] Vesperman, J., Introduction to CVS. http://www.linuxdevcenter.com/pub/a/linux/2002/01/03/cvs_intro.html.
- [11] Wang, X. D., Yang, X., Allan, R.: Flexible Grid Portlets to Access Multi Globus Toolkits. In GCC2006: CVRE06 Workshop. Changsha, China, 21-23 Oct. 2006.
- [12] Zhuge, H.: Discovery of Knowledge Flow in Science. Communications of the ACM, 49(5) (2006) 101-107.