

Bringing AJAX to Grid Portals

Xiaobo Yang
CCLRC e-Science Centre
Daresbury Laboratory
Warrington WA4 4AD, UK
x.yang@dl.ac.uk

Rob Allan
CCLRC e-Science Centre
Daresbury Laboratory
Warrington WA4 4AD, UK
r.j.allan@dl.ac.uk

ABSTRACT

Grid portals provide researchers the power of Grid through Web browsers. Today these portals are mostly Java-based and therefore the Java Portlet Specification 1.0 (JSR 168) has been widely adopted. Acting as Web components, portlets wrap relatively independent individual functionalities like Grid proxy credential management and job submission. These components can then be aggregated for construction of portal pages. One of the key issues Grid portals are facing now is that they are not as attractive as desktop applications because they lack instant response and complex visual effects. To address this, we are going to investigate Asynchronous JavaScript and XML (AJAX). AJAX is becoming increasingly popular in the Web development community, in particular, it is one of the key aspects of today's Web 2.0. In this paper, we will discuss how Grid portals can benefit from AJAX so that more end-users could be attracted.

KEYWORDS: Asynchronous JavaScript and XML, Grid, portal

1. INTRODUCTION

Grid portals are prevailing as user interfaces to the Grid technology. Based on Web technologies, they enable researchers to access remote computational and data Grid resources through a central gateway. Grid portals naturally inherit all the usual benefits of Web applications. For instance, a researcher is able to consume Grid resources with only a Web browser equipped on their desktop or mobile devices without any local installation of a Grid middleware. The portal can be upgraded without intervention of end-users. At the same time, Grid portals also face the same issues as Web applications, for example, lack of instant response. When a user submits a form by clicking a submit button, they have to wait until the response

comes back to their browser from the Web server. During this period, the user does not know what is happening behind the scenes. A typical Grid portal also lacks rich and colourful visual effects attractive to end-users. For example, when a user uploads a file, all they can see on the Web page is "Click the button submit only once, please wait patiently during file upload". The user usually does not know the upload progress until it is completed. We believe users would be more likely to use Grid portals if more visual information were provided.

Since 2004, *Web 2.0* [18, 15] has become a buzzword, although there is no precise definition but see Wikipedia for a discussion http://en.wikipedia.org/wiki/Web_2.0. According to our understanding, the core of *Web 2.0* implies two points.

1) Internet users are becoming information providers besides retrievers. Internet is no longer simply acting as a massive digital library but becoming a platform to hold data, with blogs and wikis emerging as services hosting users' contributions. This trend brings a lot of new potential usages of the Web. For example, collaboration becomes a lot more easier than before and community-driven Web applications are emerging. Two famous *Web 2.0* example sites are www.flickr.com and www.youtube.com for sharing photos and videos respectively. No matter how a *Web 2.0* application looks, personal data is becoming central of the Web. In order to understand and share this data, research activities such as semantic Web and Web mining are attracting more researchers.

2) Web applications are becoming Rich Internet Applications (RIA). In order to bring new experiences to users, Web applications today are equipped with elements for improved interfaces and, more importantly, response time. As one of the keys for implementation, AJAX combines various technologies as described by Garrett [14]:

- XHTML and CSS for presentation and accessibility;
- Document Object Model (DOM) for dynamic display and interaction;
- XMLHttpRequest for asynchronous data retrieval;
- JavaScript for binding everything together with client-side interactions.

The major contribution in AJAX is the *XMLHttpRequest* object which makes it possible to send asynchronous calls from a browser to a Web server. Once response data from the server arrives, the Web page being viewed will be dynamically updated using DOM without refreshing the whole page. A clear description of the difference between classic and AJAX Web application models is given by Garrett [14].

Google provides some very good AJAX based Web applications including its Gmail, Google Suggest and Google Maps. For example, the latter gives an end-user nearly instant response when an update of the current map is required. This application demonstrates great improvement to the user experience as compared to traditional online map sites like *www.multimap.com* which requires an explicit request/response communication cycle against the Web server each time an update of the current view is required.

In this paper, we are going to investigate how Grid portals can also benefit from AJAX. The motivation comes from the objective of attracting more Grid users. The paper is organised as follows. After a brief description of some related work, a typical example Grid portal will be described and its shortcomings discussed. AJAX will then be introduced and used to modify some of the Grid portlets in our example, demonstrating what AJAX brings. Some issues about AJAX will then be discussed before our concluding remarks.

2. RELATED WORK

The core (generic) functionalities of Grid portals today are quite mature. For example, the OGCE (Open Grid Computing Environments) Portal Toolkit [8, 5] and the GridPort Toolkit [13, 2] are two well-known toolkits aiming at providing a set of JSR 168 [4] compatible portlets for Grid portal developers. Recently GridPort has become part of OGCE which also includes contributions from the Extreme Lab, the Community Grids Lab at Indiana University, and the Sakai Project [6]. Similar to the UK National Grid Service (NGS) Portal [22], Grid access APIs are provided by the Java CoG Kit [20]. While adoption of AJAX in Grid portal development has been discussed by researchers from Indiana University and the OGCE

collaboration, currently there is no published information available on this work.

In fact adoption of AJAX in Grid portals has been discussed mainly at workshops. Jason Novotny from University of California, San Diego, USA, who led the GridSphere portal framework [17, 19, 3] development has mentioned two main usage scenarios with AJAX in Grid portals: 1) portlet developers can make use of AJAX to create more dynamic and responsive portlets; 2) support of AJAX can be done at the portal level. For example, portal frameworks like GridSphere may update an individual portlet without refreshing the entire portal page. At the time this paper was written, there were two AJAX portlets available from the GridSphere Web site – an AJAX hello world and a Google map portlet. These illustrate refreshing individual portlets without affecting other portlets in the same portal page. Considering code portability, the first approach is recommended because portlets should be deployable under various portal frameworks.

3. A TRADITIONAL PORTLET-BASED GRID PORTAL

A simplified UK NGS Portal [22] is used here as a prototype portal. The set of Java CoG based JSR 168 compliant portlets listed below has been selected for construction of this portal:

- a proxy credential manager portlet for retrieving/renewing Grid proxy credentials;
- a portlet for submitting jobs to a remote computational Grid node;
- a batch job monitor portlet;
- a GridFTP [10] based file transfer portlet for uploading, downloading files and transferring files between two remote Grid nodes (GridFTP 3rd-party transfer);
- a LDAP/ MDS [12] query portlet for retrieving information about Grid resources.

These portlets are placed within one portlet application so that Grid proxy credentials can be shared among all portlets. Suppose a researcher called John is going to use this Grid portal. A typical scenario is that John first makes use of the LDAP/ MDS query portlet to have a look at all computing/ data Grid resources registered at the MDS server. This will help him to decide which node(s) to use. He then uses the proxy manager portlet to retrieve a credential from a remote MyProxy [11] server (we assume John has already created a long-term proxy credential on the MyProxy server). After that he may use the file transfer portlet to upload his code and parameter files for

numerical simulations on the chosen remote computing node. The job submission portlet is used to start his numerical simulation, and John may submit a set of jobs with different parameters on different nodes to do his research. After job submission, he will probably make use of the batch job monitor portlet to have a look at the status of all jobs he submitted. When each job finishes, John can use the file transfer portlet again for transferring his data to a data Grid node for storage and further analysis.

Although the above cycle works fine for execution of the generic Grid tasks, some of the shortcomings encountered are:

- for each action to be taken, the user has to wait for responses returned from the server before a portal page can be refreshed;
- if two or more portlets are placed inside one portal page, all portlets need to be refreshed when an action is submitted;
- using the batch job monitor portlet, the user has to refresh job status manually by clicking a refresh button;
- when a file is uploaded or GridFTP 3rd-party file transfer is invoked, the user does not know from the file transfer portlet what progress is being made.

Shortcomings like this can be categorised as lack of instant response and visual support. Besides that, portlet developers also suffer lack of inter-portal communication support. In the next section, we will demonstrate how our investigation on AJAX in Grid portals aims to solve these shortcomings.

4. AN AJAX PORTLET-BASED GRID PORTAL

In this section, we will see how Grid portlets listed in the previous section can benefit from AJAX.

4.1. LDAP/MDS Query Portlet

In the case of the NGS Portal, the LDAP/ MDS query portlet has nothing to do with other portlets since there is no need to make use of the proxy credential as the MDS server is publically accessible. For further LDAP/ MDS queries, AJAX is used so that this portlet can update itself without affecting other portlets viewed in the same portal page. To do this we make use of DWR (Direct Web Remoting) [1] to bring AJAX support in our portlets. DWR is ideal for Java developers (in our case, portlet developers) because it allows JavaScript in a Web browser to interact with Java objects on the server side.

A screenshot of the LDAP/ MDS query portlet, together with the two batch job monitor portlets to be discussed later, is illustrated in Fig. 1. The main output of the LDAP browser shown in Fig. 1 is between the two horizontal lines within which each icon and the text near the folder icon is a link for further query. For example, when a user clicks the folder icon left of "*Mds-Host-hn=Grid-compute.leeds.ac.uk*", a query will be sent via the portal server to the LDAP server to ask for information about its child nodes; while "*Mds-Host-hn=Grid-compute.leeds.ac.uk*" itself represents a further query for its attributes.

Rather than submitting new action requests to the portlet, these links have been changed to invoke a server side object for LDAP/ MDS query using DWR. Therefore when a new query result is due, the portlet will be updated without affecting others in this portal page. Also in practice, AJAX makes this portlet more responsive as users get nearly instant response from the portal server.

4.2. Proxy Manager And Job Submission Portlet

The proxy manager and job submission portlet are left unchanged. This is because when the proxy manager portlet retrieves a new or updates a current credential, other Grid-related portlets are required to know this change. By putting all portlets inside one portlet application, we make use of the *PortletSession* for sharing this Grid proxy credential. The same thing happens to the job submission portlet. When a job is submitted by this portlet, the batch job monitor should be notified that a new job has been submitted.

4.3. Batch Job Monitor Portlet

Two tests have been performed on the batch job monitor portlet. Firstly a timer was added to this portlet so that a request will be periodically sent to the portlet. Because output of a portlet is only an HTML fragment, it does not have access to the *onload* parameter of the *body* tag. Therefore we simulate a new action request using JavaScript to the portlet after some time interval.

Another test has been performed to study inter-portal communication using AJAX. In this test, the batch job monitor portlet has been split into two portlets. One of them is responsible for listing all jobs submitted by the current user. When a job is selected in this portlet, details of this job will be displayed inside the other portlet.

In Fig. 1, a screenshot of these two batch job monitor portlets plus the LDAP browser portlet are depicted. The

left one called "Batch Job Monitor All Jobs Portlet" is responsible for listing all jobs submitted by the current user. The right one, "Batch Job Monitor One Job Portlet", is designed to display detailed information about a job selected in the "Batch Job Monitor All Jobs Portlet". In Fig. 1, information about job 1 is shown.

4.4. File Transfer Portlet

The file transfer portlet has been designed to perform tasks including: 1) file upload from user's local computer; 2) file download from a remote Grid node; and 3) 3rd-party file transfer (between two remote hosts). GridFTP [10] is a high-performance, secure, reliable data transfer protocol optimised for high-bandwidth wide-area networks and is the underlying protocol for realising these tasks invoked from the portal server. When a user wants to upload a file or manage 3rd-party transfer of a large file, the portlet simply executes the request without any notice given to the user of progress. Two modifications have therefore been applied to this portlet:

- an upload progress bar is added so that during file upload user knows the status;
- during GridFTP 3rd-party file transfer, the current file name and progress are displayed so that user gets an idea about the transfer status. This is especially useful for any time-costly transfer, for instance one or more large files or transfer of whole directories.

Figs. 2 and 3 display two screenshots of a file being uploaded and a GridFTP 3rd-party file transfer respectively. Eighty three percent of the uploaded file has been transferred in Figs. 2 and 43% of a file called *junk_file_1000MB* has been transferred from *Grid-compute.oesc.ox.ac.uk* to *Grid-data.rl.ac.uk*.

5. FURTHER DISCUSSIONS

In the above section, we have demonstrated how Grid portals can immediately benefit from AJAX. In general, AJAX enables us to solve shortcomings listed in Section 3 and therefore brings new experiences to Grid portal users: 1) Grid portal page is now more responsive than before; 2) visual effects like file upload progress bar can be added. We have also evaluated AJAX to link portlets together so that inter-portlet communication can be realised.

5.1. Photo Show And Chat

We have performed further investigations with AJAX to bring additional collaborative elements to Grid portals. For example, we have developed two other portlets including a photo show and a chat portlet for demonstration purpose. The former periodically displays a set of shared pictures

one by one in a portal page, the latter enables synchronous communications between on-line portal users. Fig. 4 gives a screenshot of these two portlets displayed together.

AJAX can bring a lot of traffic to the portal server. For example, the photo show and chat portlets both require sending requests to the portal periodically. When a lot of users are accessing these portlets, performance will become a big concern. Therefore besides avoidance of unnecessary calls to the portal server, techniques like portal server clustering should be considered in that case. This is likely to become more often the case with the growth of on-line collaboration tools based on Web 2.0. Sakai has already addressed a number of these issues as it is used in large educational establishments as an open-source Collaborative Learning Environment [6].

5.2. Bringing Web 2.0 Elements in Grid Portals vs. Bringing Grid to Web 2.0 Applications

We have demonstrated the approach and advantages of bringing Web 2.0 elements in a Grid portal. Grid portlet developers can always improve portlets like the chat portlet to make them good tools for portal users to communicate with each other. The same objective can be achieved for other tools like blog and wiki. In fact, there exists another approach, i.e., bringing the Grid to Web 2.0 applications. This arises from our experience in the UK JISC (The Joint Information System Committee) funded Sakai VRE Demonstration Project [9]. Sakai [6] is targeting the higher education market and has some Web 2.0 elements including blog and wiki tools and inevitably has many other collaborative tools like chat, discussion, and online presentation. If we bring Grid functionalities to Sakai, then Sakai can be treated as a fully-functional Grid portal. While support of JSR 168 [4] is still being developed in Sakai, Grid portlets are accessible to Sakai users through the Web Services for Remote Portlet (WSRP) [7] approach. In brief, WSRP makes use of the Web service concept to publish portlets as remote portlets through a WSRP producer (Web service). Thus any application equipped with a consumer (Web service client) is able to consume these portlets without local deployment. For detailed information about this approach, refer to [23].

Whilst WSRP is a promising approach for integration of existing Grid portlets inside Web 2.0 applications, there exist issues about the WSRP 1.0 specification itself and its implementations. In [21], some of the problems have been discussed based on our study of WSRP4J, the Apache implementation of WSRP 1.0. Furthermore, when we talk about AJAX in this paper, remote calls from Web browsers

to the portal server are handled behind the scenes. Unfortunately, in the WSRP scenario these remote calls will not be able to be completed because calls are targeting the WSRP producer which is typically different from the portal server. Such a call is not allowed in Firefox owing to security concerns, although Internet Explorer does allow users to connect to a remote host other than the portal server if they insist to do so. To solve this issue, a proxy service architecture or enterprise service bus (ESB) as discussed in [16] are potential choices. In [23] the proxy approach has been successfully tested using AJAX with WSRP. Further investigations of high-level AJAX toolkits like DWR in the WSRP scenario are required.

6. CONCLUSIONS

In this paper, we have demonstrated how Grid portals can benefit from AJAX. In conclusion, AJAX removes some shortcomings current Grid portals are facing. In particular, AJAX brings Grid portal users new experiences. For example, AJAX makes it possible to refresh an individual portlet without affecting other portlets inside the same portal page. This behaviour also makes Grid portals more responsive in practice. With the help of AJAX, visual effects can be easily added to Grid portlets. In this paper, we experimented with a file upload progress bar and GridFTP 3rd-party file transfer status for reporting progress to end-users. We have also demonstrated the power of AJAX by realising inter-portlet communications and bringing Web 2.0 collaborative elements to Grid portals.

Whilst adoption of AJAX in Grid portlet development as we have done seems to be a good approach for improving a Grid portal's usability, integration of these AJAX-enabled Grid portlets encounters security issues when deployed in a remote portlet. This for instance happens when we try to integrate them inside Sakai alongside Web 2.0 tools using WSRP. Further investigation is required to see how this problem can be solved. Hopefully JSR 286 and WSRP 2.0, successors of JSR 168 and WSRP 1.0 respectively, will further highlight the usage of AJAX for portlet developers. Support of AJAX at the portal framework level should also be standardised so that portlet developers do not need to worry about tasks like refreshing of an individual portlet and moving portlets inside a portal page.

ACKNOWLEDGEMENT

We thank the anonymous reviewers for their insightful comments which have helped to improve this paper. This work was undertaken at the CCLRC e-Science Centre,

Daresbury Laboratory supported by UK JISC (The Joint Information System Committee).

REFERENCES

- [1] DWR: Direct Web Remoting, <http://getahead.ltd.uk/dwr/>.
- [2] GridPort4, <http://Gridport.net/>.
- [3] GridSphere, <http://www.Gridsphere.org/>.
- [4] JSR 168: Java Portlet Specification 1.0, <http://www.jcp.org/aboutJava/communityprocess/final/jsr168/>.
- [5] OGCE: Open Grid Computing Environments, <http://www.collab-ogce.org/ogce2/>.
- [6] Sakai: Collaboration and Learning Environment for Education, <http://sakaiproject.org/>.
- [7] WSRP 1.0: Web Services for Remote Portlets, <http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf>.
- [8] Alameda, J., M. Christie, G. Fox, J. Futrelle, D. Gannon, M. Hategan, G. von Laszewsk, M.A. Nacar, M. Pierce, E. Roberts, C. Severance, and M. Thomas, "The Open Grid Computing Environments Collaboration: Portlets and Services for Science Gateways," <http://doi.wiley.com/10.1002/cpe.1078>, October 2006.
- [9] Allan, R., D. Chohan, X. Wang, X. Yang, R. Crouchley, A. Fish, M. Gonzalez, M. Baker, H. Ong, M. Dovey, and F. Pinto, "Virtual Research Environments: Sakai Demonstrator," Proc. UK e-Science All Hands Meeting 2005, available on CDROM, Nottingham, UK, September 2005, pp. 208-215.
- [10] Allcock, W., J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, and I. Foster, "The Globus Striped GridFTP Framework and Server," Proc. SC05, Seattle, USA, November 2005.
- [11] Basney, J., M. Humphrey, and V. Welch, "The MyProxy Online Credential Repository," *Softw.: Pract. Exper.*, Vol. 35, No. 9, 2005, pp. 801-816.
- [12] Czajkowski, K., S. Fitzgerald, I. Foster, and C. Kesselman, "Grid Information Services for Distributed Resource Sharing," Proc. HPDC-10, IEEE Computer Society, San Francisco, CA, USA, August 2001.
- [13] Dahan, M., M. Thomas, E. Roberts, A. Seth, T. Urban, D. Walling, and J.R. Boisseau, 'Grid Portal Toolkit 3.0 (Grid-Port)', Proc. HPDC-13, IEEE Computer Society, Honolulu, Hawaii, USA, June 2004, pp. 272-273.
- [14] Garrett, J.J., "AJAX: A New Approach to Web Applications," <http://www.adaptivepath.com/publications/essays/archives/000385.php>, February 2005.

- [15] Graham, P., “Web 2.0”, <http://www.paulgraham.com/Web20.html>, November 2005.
- [16] Margaglione, J., “AJAX programming in BEA WebLogic Portal 8.1, part 1 & 2,” <http://dev2dev.bea.com/pub/a/2006/01/ajax-portal-1.html>, <http://dev2dev.bea.com/pub/a/2006/03/ajax-portal-2.html>, April 2006.
- [17] Novotny, G., M. Russell, and O. Wehrens, “GridSphere: a Portal Framework for Building Collaborations,” *Concurrency: Pract. Exper.*, Vol. 16, No. 5, 2004, pp. 503-513.
- [18] O’Reilly, T., “What Is Web 2.0”, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-Web-20.html>, September 2005.
- [19] Russel, M., J. Novotny, and O. Wehrens, “The Grid Portlets Web Application: A Grid Portal Framework,” Proc. PPAM 2005, Poznan, Poland, September 2005, pp. 691-698.
- [20] von Laszewski, G., I. Foster, J. Gawor, and P. Lane, “A Java Commodity Grid Kit,” *Concurrency Computat.: Pract. Exper.*, Vol. 13, No. 8-9, 2001, pp. 643-662.
- [21] Yang, X., and R. Allan, “A Deep Look at Web Services for Remote Portlets (WSRP) and WSRP4J,” GCE’06: Second International Workshop on Grid Computing Environments, held in conjunction with SC06, Tampa, FL, USA, November 2006.
- [22] Yang, X., D. Chohan, X. D. Wang, and R. Allan, “A Web Portal for the National Grid Service,” Proc. UK e-Science All Hands Meeting 2005, available on CDROM, Nottingham, UK, September 2005, pp. 1156-1162.
- [23] Yang, X., X. D. Wang, R. Allan, M. Dovey, M. Baker, R. Crouchley, A. Fish, M. Gonzalez, and T. van Ark, “Integration of Existing Grid Tools in Sakai VRE,” International Workshop on Collaborative Virtual Research Environments (CVRE06), Proc. Fifth International Conference on Grid and Cooperative Computing Workshops (GCCW2006), IEEE Computer Society, Changsha, China, October 2006.

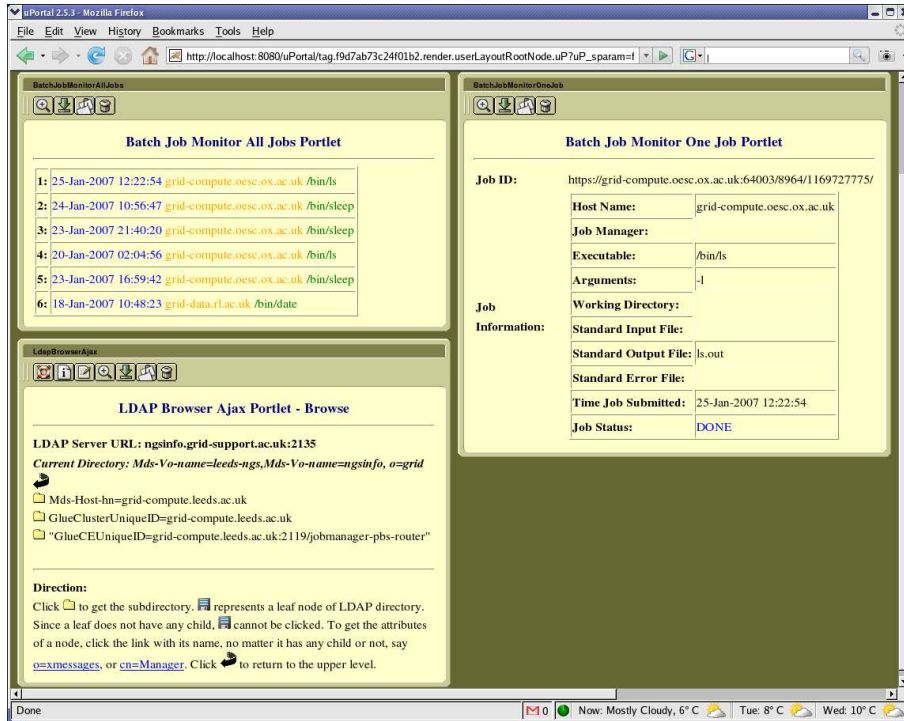


Figure 1. AJAX Enabled Portlets – LDAP/ MDS Query and Batch Job Monitor

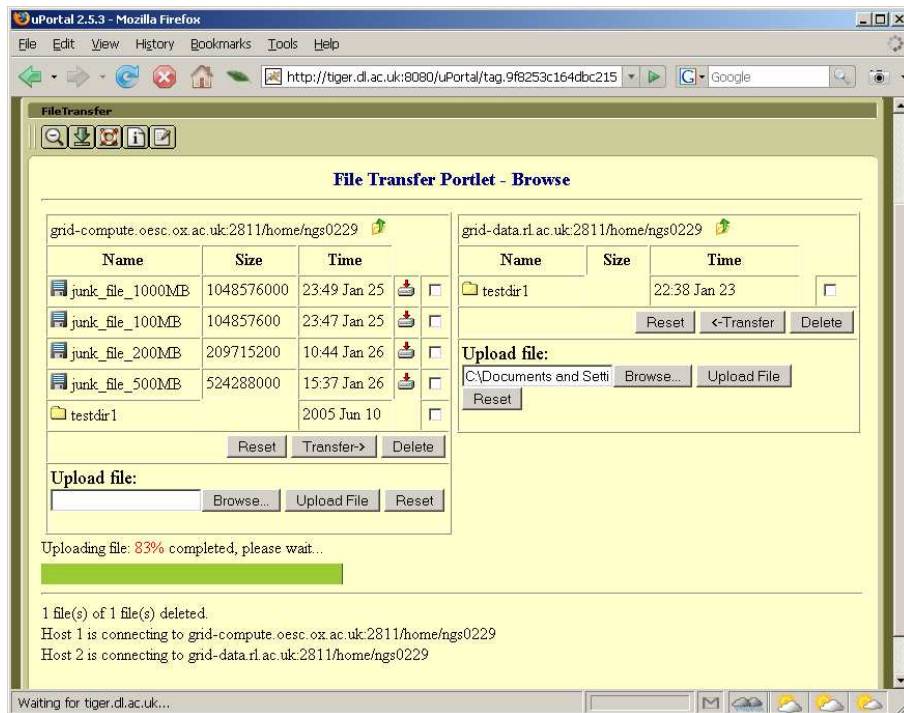


Figure 2. AJAX Enabled File Transfer Portlet – File Upload

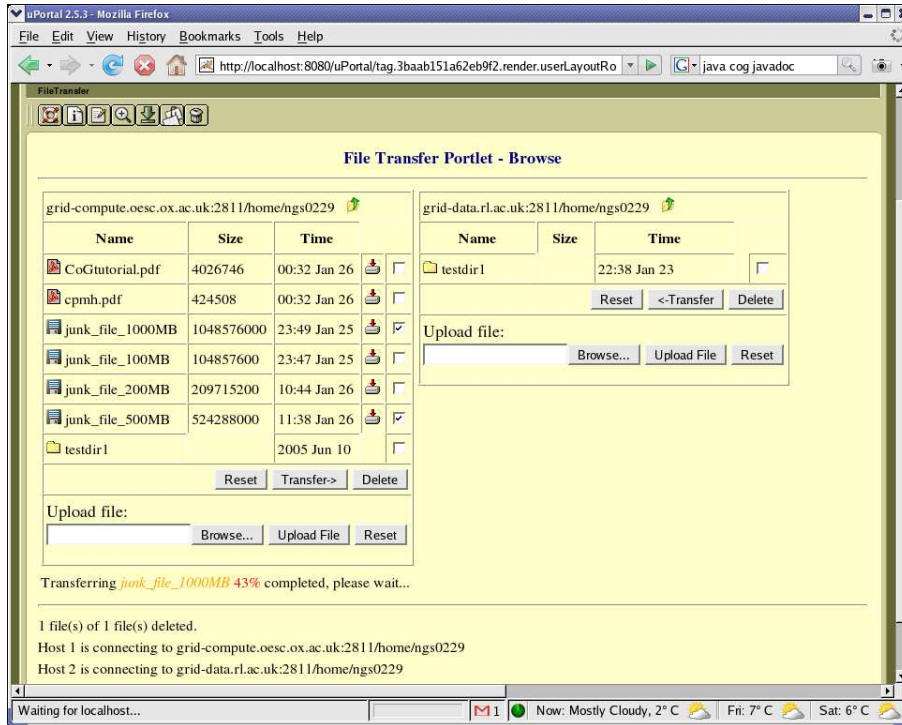


Figure 3. AJAX Enabled File Transfer Portlet – GridFTP 3rd-party Transfer

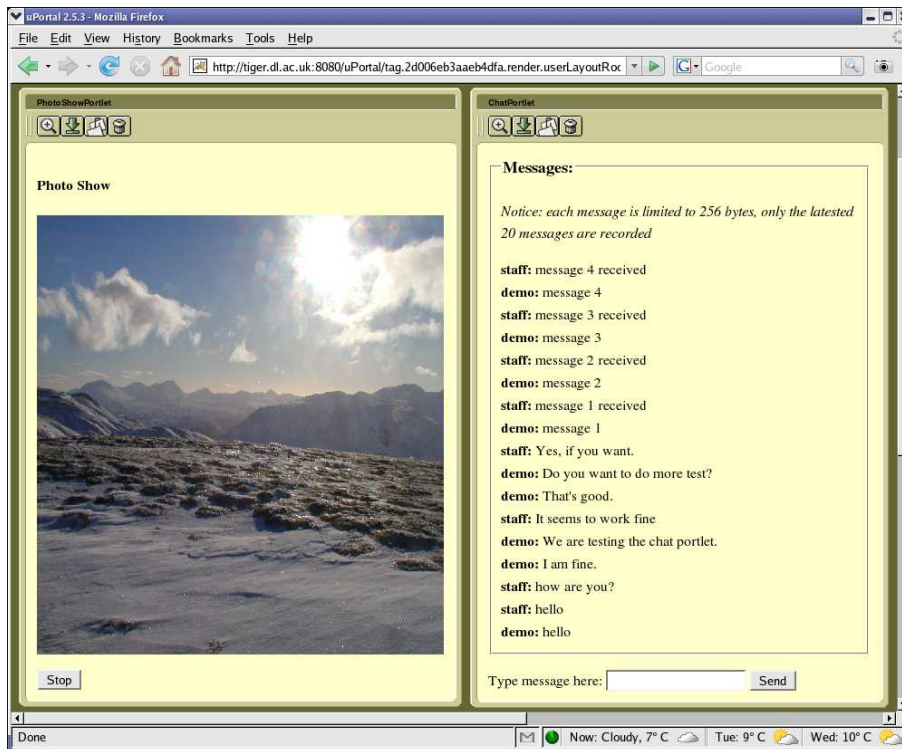


Figure 4. AJAX-based Portlets – Photo Show and Chat