

SRB Data Resources in Computational Grid Workflows

Tamas Kiss, Alexandru Tudose, Peter Kacsuk, Gabor Terstyanszky

Centre for Parallel Computing, University of Westminster
115 New Cavendish Street, London, W1W 6UW
e-mail: gemplca-discuss@cpc.wmin.ac.uk

Abstract

Besides computation intensive tasks, the Grid also facilitates the sharing and processing of very large databases that are distributed over multiple resources and administrative domains. There are several lower level tools that manage and provide access to distributed databases and file systems. However, these tools offer only low level interfaces, or limited to browsing and transferring the data between Grid sites. This paper describes how data can be fed into and utilised in data driven computational workflows. The workflow engine of the P-GRADE Grid portal is used as an implementation platform to be integrated with different Grid data solutions. As an example, the paper describes how the portal and the workflow engine have been integrated with the SDSC Storage Resource Broker (SRB) in order to utilise SRB data resources as both inputs and outputs of workflow components. The solution automates data interaction in computational workflows allowing users to seamlessly access and process data stored in SRB resources.

1. Introduction

Many scientific and business applications require access to large, distributed data collections. These data collections can be highly heterogeneous spanning several administrative domains. In order to efficiently handle this large amount of distributed data, an abstraction has to be provided to the user that hides the multiplicity of platforms, resources and data types, as well as provides the users with seamless access to data stored within the collections.

Current production Grids all aim to support data intensive applications and offer solutions such as OGSA-DAI [1] (Open Grid Services Architecture Data Access and Integration) or SRB [2] (SDSC Storage Resource Broker) to access large data collections. These tools provide the required abstraction of data and support interoperation of heterogeneous data sources.

Despite the useful functionality provided, most end-users find it difficult to access these relatively low-level tools directly. End-users require a graphical user interface, typically a Grid portal, where they can create, execute and monitor their applications in an integrated environment. Besides simple job execution some Grid portals also offer additional functionalities such as workflow composition, or easy selection and execution of

applications from code repositories with custom input parameter values. These applications may require access to large data collections, as described previously.

Unfortunately, current Grid portals and application hosting environments provide very little or no support at all to access advanced data manipulation tools, such as OGSA-DAI or SRB. Current solutions are limited to specific browser portlets that enable accessing data collections and doing the necessary data movements and transformations prior or after job execution manually. In case of more complex application scenarios, that require the execution of several jobs with complex dependencies between them, this manual data manipulation is not feasible. On the other hand, workflow solutions, such as Triana [3], Taverna [4] or the P-GRADE portal [5] workflow engine, provide automatic orchestration of these complex scenarios, staging and automatically transferring data along the workflow. However, these solutions are currently limited concerning their data manipulation capabilities and the data sources they can utilise.

The aim of this paper is to describe how computational workflows can be extended with more sophisticated data management capabilities. In order to demonstrate this concept the workflow engine of the P-GRADE grid portal has been extended to seamlessly access data stored in SRB

repositories. Section 2 of this paper gives an overview of how current Grid portal and workflow solutions support data access and manipulation. Section 3 describes how data is currently handled by the P-GRADE portal, followed by the description of the SRB/P-GRADE integration in section 4. Finally, future work and conclusions are given in section 5.

2. Related work

There are several research efforts and existing tools that support accessing Grid resources and creating computational workflows within a high-level user-friendly environment. These tools all provide means of executing standalone applications or orchestrating more complex workflows, and as such they all have data management capabilities too. However, the type of data resources as input or output to applications or workflow components are either very restricted, or they provide only basic operations and limited or no integration at all with the workflow engine. In this section we give a brief overview of Grid portal and workflow management tools concentrating on their data managements capabilities.

First, a few Grid-portal solutions are introduced providing data, and more specifically, SRB support. As there are many different solutions in this area, our aim was not to give a full list rather to highlight the most relevant and typical solutions.

The most relevant solution for the UK e-Science community is the NGS-portal [15]. The latest version of this portal is an application repository where all necessary information required to execute hosted applications is captured in full by their associated JSDL (Job Submission Description Language) [22]. Users can upload and share JSDL descriptions, and submit and monitor jobs selected from the repository and mapped to NGS resources. The current version of the portal only supports data staging from GSIFTP servers. However, SRB support is also in the development pipeline and a previous version of the NGS-portal has already included an SRB Browser portlet [16].

The BIRN (Biomedical Informatics Research Network) Portal [17] provides BIRN participants with a single sign-on web portal to access Grid resources and data grid files. It is using SRB as distributed data manager. Regarding the SRB-client portlet, the BIRN portal allows the users to navigate, upload and download files, operate permission changes on the files they own, and

view and modify metadata associated with their files. Shared data spaces for collaborative projects can also be created on the fly [18].

The NCMIR (National Center for Microscopy and Imaging Research) develops 3D imaging and analysis technologies to help biomedical researchers understand biological structure and function relationships in cells and tissues. The NCMIR portal provides their users with SRB client capabilities [19], which are made available through a portlet based on JARGON, the java API provided to SRB. This portlet provides similar functionality to the BIRN portal including collection browsing, file upload and download, permission changes and metadata modification.

Although all above solutions provide SRB support, they are all limited to simple or more sophisticated SRB browsing. As none of these portals have workflow support capabilities they can neither provide workflow level data or SRB support. On the other hand there are widely used workflow tools in Grid systems that stage data automatically through complex application scenarios. Unfortunately, the type of data sources these workflows can utilise is limited.

One prominent example for a workflow engine is Taverna, developed by the myGrid project [20]. Taverna [21] is a graphical workbench tool for creating and running workflows composed of primarily Web services. It was originally developed for biologists and bioinformaticians but it is now also used by a growing community from other fields such as chemoinformatics, astronomy and health informatics. The aim of the tool is to put together publicly available (Web) services, transfer data between them and orchestrate their execution. Taverna is a very effective workflow management environment but not easily adaptable for current production Grids. Its main aim is not job submission rather the connection and orchestration of pre-deployed services. The data input is defined by the implementation of the actual service.

Another example is Triana, a problem solving environment developed at Cardiff University [3]. Triana is a graphical environment that allows the creation of powerful applications with a minimum user effort and no programming. Using Triana, programs can be assembled from a set of building-blocks by dragging them into a work-space window and connecting them. However, Triana is not integrated with Grid data solutions like SRB.

3. Data Driven workflows in the P-GRADE portal

The P-GRADE portal is a general purpose, workflow oriented computational Grid portal that supports the development and execution of workflow-based Grid applications. The portal has been deployed as production level service portal on several production Grids, including the UK National Grid Service [6], numerous virtual organisations of the EGEE Grid (GILDA, VOCE, SEE-Grid etc.) [7], the TeraGrid [8], the Open Science Grid [9] and several national Grids (HunGrid, Croatian Grid, Turkish Grid etc.). Detailed description of the portal can be found in several publications such as [5], or on the P-GRADE portal Website [10]. Here we only concentrate on describing the current data manipulation and transfer capabilities of the portal in order to explain the motivations to our work.

• A directed acyclic graph where:

- Nodes represent jobs - either sequential or parallel programs
- Ports represent input/output files the jobs expect/produce
- Arcs represent file transfer between the jobs

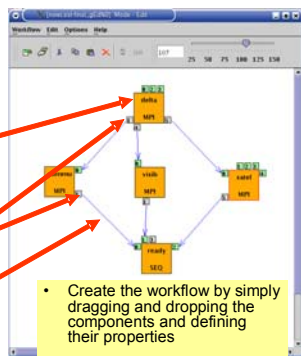


Figure 1 – A P-GRADE portal workflow

The basic unit of execution in the P-GRADE portal is the workflow. A P-GRADE portal workflow is a directed acyclic graph where the nodes of the graph represent sequential or parallel jobs, as illustrated on Figure 1. The nodes are communicating with each other via the means of file transfer. Each job requires zero or more input files and produces zero or more output files. The files are currently transferred between the nodes via GridFTP file transfer, represented by the arcs of the graph on Figure 1. The arcs are connecting the output ports of one job to the input ports of another.

When creating a computational workflow the user either uploads the executables to the portal server from a local machine, or selects the codes from a legacy code repository called GEMLCA (Grid Execution Management for Legacy Code Applications) [11]. In both cases the input files to the workflow can currently be either local or

remote input files. Local input files are uploaded from the user's local file system to the portal server during workflow creation, and from here at execution time they are transferred to the Grid compute resource. Local input files are typically small in size. However, a Grid application may require huge amount of input data that is stored in external servers. In this case the user can define the input as "external" and provide its GridFTP address (in case of Globus based Grids) or LFC logical file name (in case of EGEE type of Grids). An external input file is always directly transferred from its storage location to the executor site.

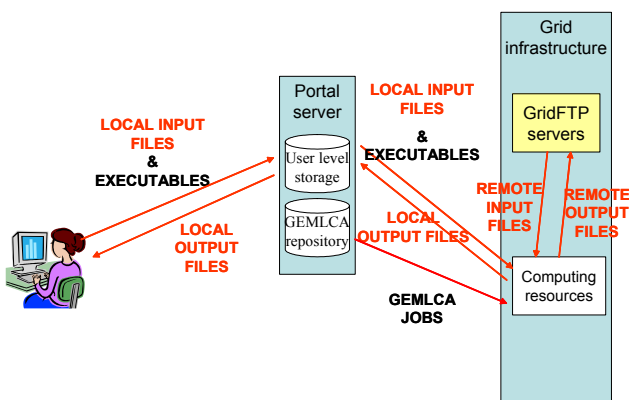


Figure 2 – Local and remote input/output files and executables

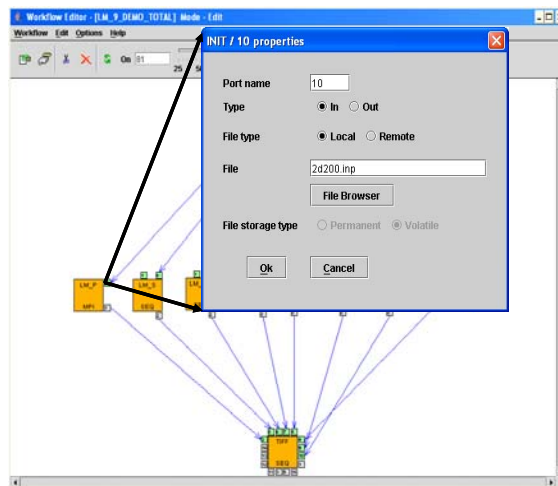


Figure 3 – Input/output file definition in the P-GRADE portal

Similarly to input files, output files can also be either local or remote, following the previously

described logic. In case of local output files the user can define the output as permanent that will be available for download on the portal server after workflow execution, or volatile that will only be used for internal communication and will automatically be deleted when the workflow has finished. Remote output files are always permanent.

In order to move remote and local input/output files the portal extends the job executable with a pre- and a post-script. The pre-script is responsible for transferring the input file to the compute resource before the execution of the job either from the portal machine (local input files) or from a remote storage resource (remote input files). On the other hand, the post-script transfers the results back to the portal server (local output files) or to a remote storage location (remote output files).

Figure 2 illustrates how executables and local and remote input/output files are transferred by the P-GRADE portal. Figure 3 shows the current dialog box to define input/output files of a job highlighting the options detailed above. Finally, figure 4 illustrates how pre-and post-scripts are used to move input/output files. The aim of our current research is to significantly extend these options by seamlessly integrating more sophisticated data resources, such as SRB repositories or databases exposed by OGSA-DAI to P-GRADE portal workflows.

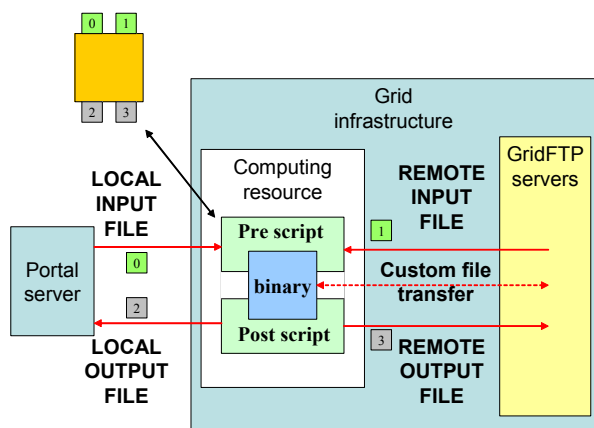


Figure 4 – Pre- and post-script for file transfer

4. SRB Data Resources in P-GRADE portal workflows

This section describes how the data handling capabilities of the P-GRADE portal have been

extended by adding SRB access. First, different alternative approaches towards SRB integration are presented and elaborated. This is followed by the description of work allowing the usage of SRB resources in input/output ports of P-GRADE workflow components.

4.1 SRB in the P-GRADE portal

The SDSC Storage Resource Broker [13] is a software product developed by the San Diego Supercomputing Centre that provides access to files and database objects seamlessly across a distributed environment. SRB abstracts the physical location of the data and the way the data is stored by presenting the user with a single file hierarchy for data that is actually distributed across multiple storage systems. SRB provides a way to access data sets and resources based on their logical names or attributes rather than their names and physical locations. The solution is widely deployed in production Grid systems including the UK NGS.

The P-GRADE Grid portal is one of the service portals operated on the UK National Grid Service (besides other large production Grids). Furthermore, a large number of NGS users utilise SRB. Therefore, the integration of SRB access capabilities to the P-GRADE portal is important and interesting to a potentially large user community.

SRB functionalities can be offered in the P-GRADE portal in different ways. The most obvious solution is to extend the portal with an SRB browser portlet. This SRB portlet can help users create, save and delete files on SRB servers, as well as to manage collections and to perform backup, replication and check-summing operations. As mentioned in Section 2, there are several JSR 168 [12] compliant portlets offering basic or more advanced SRB functionalities. Work is currently being undertaken to evaluate, compare and extend these solutions if necessary, and integrate an appropriate SRB browser portlet into the P-GRADE portal.

Besides an SRB browser portlet, the major interest of our work focuses on integrating SRB at workflow level. This integration can be done at port or at job level.

The port level integration means that input or output ports of a P-GRADE portal job could refer to files stored on SRB storage facilities and consequently, utilise SRB data stores for file storage and file retrieval. The current types of ports, “local” and “remote”, are extended with a third port type that refers to SRB resources. The

solution allows users to process input data in workflow jobs stored in SRB data stores, and also to write the output of the jobs back to an SRB resource.

The second option is the job level integration. Portal users can currently define and execute “standard” or “GEMMLCA” jobs. A standard job represents a GT2 or g-Lite job submission, while a GEMMLCA job is a service invocation that refers to a code stored in the GEMMLCA repository. This framework can be extended with a third job type, called “SRB” job. The “SRB” job is a set of SRB commands, potentially created with the help of a built in GUI within the portal, which manipulates an SRB data collection. This job can still be connected to other “standard”, “GEMMLCA” or “SRB” jobs with the help of “local”, “remote” or “SRB” ports.

The second half of this section describes the port level integration, while the job level solution will be introduced in another paper.

4.2 SRB resources as input/output ports

The port level integration of SRB into the P-GRADE portal raises three different challenges:

- the SRB client environment has to be properly set up and configured,
- SRB port definition has to be enabled in the workflow editor of the portal,
- and finally, data has to be retrieved from SRB resources before job execution, and transferred back afterwards.

4.2.1 Setting up the SRB client environment

When dealing with SRB environment descriptions, the aim was to enable access to multiple SRB servers at the same time. These SRB servers can be located in different Grids and may require different certificates to access. The P-GRADE portal is a multi-Grid portal [14] that allows users to map jobs and access data located in different production Grids (which may use different Grid middleware solutions and require different certificates) within the same workflow. The portal allows storing multiple certificate proxies on the portal server at the same time mapping them to different Grids. Our aim was to extend this multi-Grid capability to SRB resources too. In order to understand the rationale behind the selected and implemented approach, a short overview of SRB environment definition is given next.

The Storage Resource Broker comes as a client-server based middleware. While the server provides the necessary means to build collections, as well as to manage, access and preserve data in a distributed data Grid infrastructure, the client becomes the interface between the user and the data manager.

Before running any SRB client, the client environmental variables need to be set up. Usually, this is accomplished via a configuration file named *MdasEnv*. This is a simple text file that contains lines of parameter/value pairs where each value is given in quotes.

After setting up the environment variables the client can connect the SRB server and initiate communication with it. From this point further, users can access all the SRB capabilities, such as: read and write operations, building and handling collections of files, file check-summing, replication operations and backup operations as well as fine grained control of parallel file transfers.

In order to set up the client environmental variables, an SRB Settings portlet has been developed and added to the P-GRADE portal to allow users uploading their own SRB environmental files (*MdasEnv*). Here, users can load *MdasEnv* files from their own file-system, can view and modify existing SRB environmental files and most importantly, can create new ones on the fly. This portlet has been designed and implemented in such way that enables the portal to handle multiple *MdasEnv* files for a particular user. These environmental files are mapped to Grids and this way linked to potentially different Grid user certificates. The solution extends the multi-Grid capabilities of the portal to SRB resources and allows users to connect to multiple SRB servers concurrently, independently of their Grid membership.

Figure 5 illustrates an example setup where four different *MdasEnv* files are loaded to the portal, as shown in the left window of the screen. The right window displays the values of the highlighted environmental file that is mapped to the “NGS” Grid. As the portal already associates this Grid with a particular user certificate, it will use that particular certificate to access the specified SRB resource. Using the graphical user interface, users can also easily edit the content of the *MdasEnv* files without learning its actual syntax.

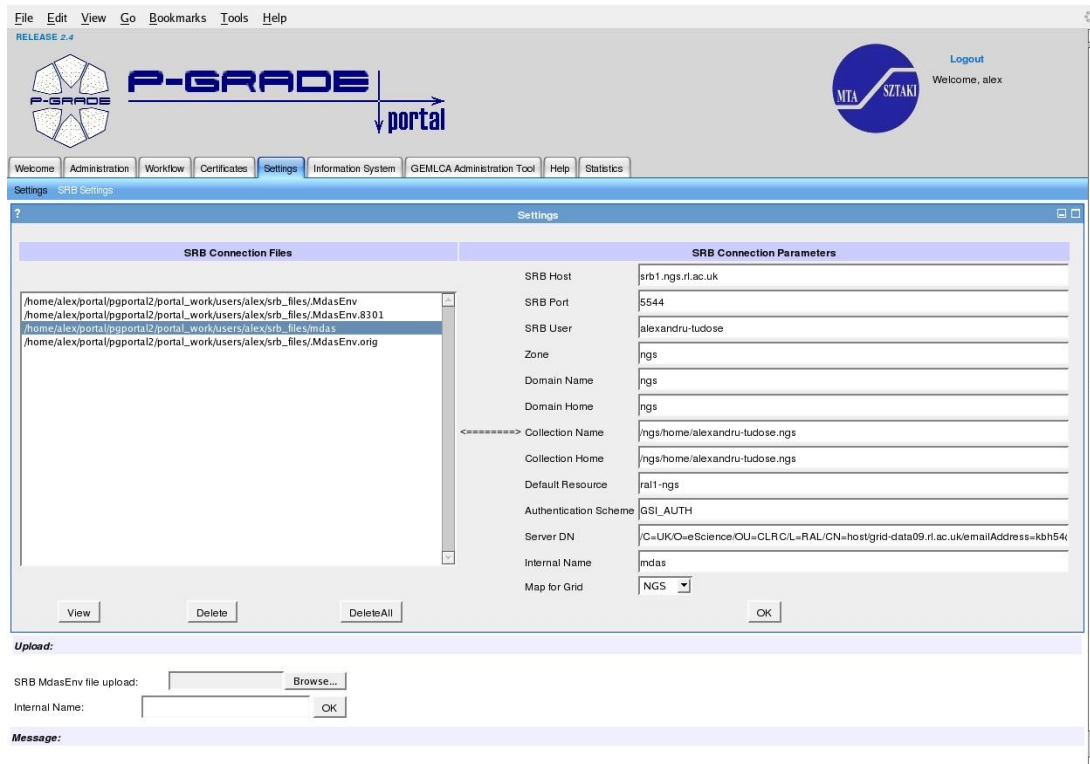


Figure 5 SRB Settings Portlet in the P-GRADE Portal

4.2.2 Workflow creation and execution using SRB ports

The next challenge after setting up the SRB environment was to modify the workflow editor of the portal to enable the creation of SRB input and output ports. This required the introduction of a new port type called “SRB” besides the currently existing “Local” and “Remote” types.

Figure 6 depicts the port properties window, modified to handle SRB file type. The user first selects the environmental file to be used in the “SRB connection file” drop down box. As this file is mapped to a particular Grid, the portal knows which certificate to use when accessing the selected collection, as it was explained in the previous section. Following this, the SRB file path is either given directly or selected by using the built in SRB file browser. Here we note that the current implementation only allows direct file path definition and the SRB browser integration is work in progress.

The final challenge is to actually access the SRB data collection to retrieve the selected files before job execution, and also to copy SRB output files back to the SRB resource. In its current

implementation, the P-GRADE portal employs a script-based solution for file staging (see figure 4). These scripts suffered necessary modifications to extend this file staging to SRB resources.

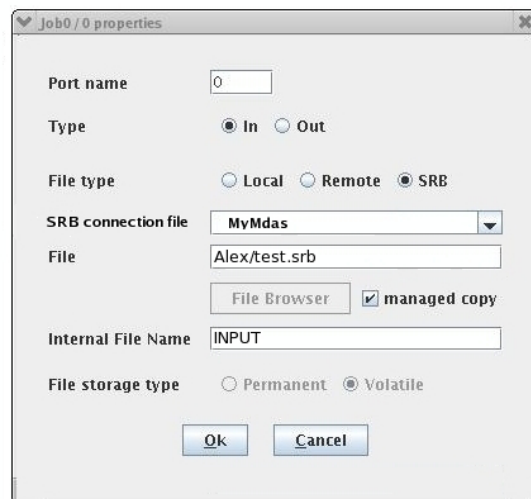


Figure 6 Port properties window extended with “SRB” port type

SRB is accessible from various client solutions such as Scomands (UNIX-like command-line

interface), Jargon (JAVA API), C client (C API), inQ (Windows based browser/query tool) and mySRB (web-based browser/query tool). After a critical analysis of SRB client solutions, the SRB command-line client (s-commands) was selected and installed on the portal server. The grounds of this decision were completeness of the command line client giving access to all the SRB functionality, and also its suitability to be integrated with the P-GRADE portal scripts.

The robustness of the solution has been addressed too. The portal machine can easily become a bottleneck if all SRB file transfers occur and terminate from this machine. In case of multiple SRB transfers or large file transfers this becomes a drawback and it could have serious impacts on the overall portal functionality and performance. The solution is to utilise direct file transfer between the SRB resource and the executor site whenever it is possible. If an executor site is registered by the portal as "SRB enabled" - indicating that it has an SRB client installed - then the portal utilises this client to directly transfer input/output files. If the executor site does not support SRB or the direct transfer failed, then the file is first transferred to the portal server utilising the portal's SRB client and then to the executor site by the means of GridFTP. These options are shown on figure 7.

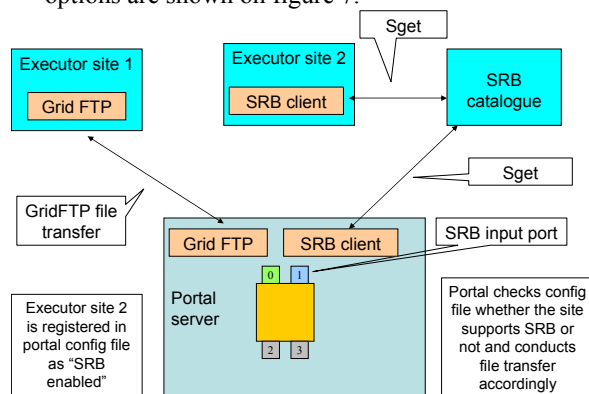


Figure 7 Direct and indirect file transfer between SRB resource and executor site

At the moment basic file staging operations are allowed from SRB ports including reading from the SRB storage facilities (corresponding to workflow INPUT ports), writing to SRB storage facilities (corresponding to workflow OUTPUT ports) and the check-summing operation to detect any corruption of files which may occur during transfers. These functionalities will significantly be extended in future releases.

5. Conclusions and Further Work

The P-GRADE Grid portal is a high-level integrated application hosting environment that assists the user-friendly creation and execution of workflow-based Grid applications spanning multiple Grids. However, the current portal only supports the access to local files or GridFTP file collections as inputs and outputs of workflow components. The aim of our research is to extend these capabilities to other widely used distributed Grid data management solutions. As a first reference implementation the P-GRADE portal workflow engine has been extended with SRB support capabilities. SRB is one of the most widely used distributed data management systems by the Grid community, and this way the integration is potentially a huge interest to both SRB and P-GRADE portal user groups. The current solution allows the definition of any input or output port of a P-GRADE portal job as either GridFTP or SRB ports. These ports can be mixed in any combination for a particular job.

Future work is two folded. On one hand other Grid data management solutions, such as OGSA-DAI are analysed and considered for integration. On the other hand, the capabilities of the current SRB integration are extended. An SRB browser is being built into the workflow editor, the group of SRB command to be utilised in ports are being extended, and a separate SRB portlet is also added to the portal.

Acknowledgement

This research work is carried out under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265).

References

- [1] Mario Antonioletti et. al.: The design and implementation of Grid database services in OGSA-DAI, Concurrency and Computation: Practice and Experience, Volume 17, Issue 2-4, Pages 357 – 376, Special Issue: Grids and Web Services for e-Science, 2005 John Wiley & Sons, Ltd.
- [2] Arcot Rajasekar et. al. Storage Resource Broker - Managing Distributed Data in a Grid, Computer Society of India Journal, Special Issue on SAN, Vol. 33, No. 4, pp. 42-54, Oct 2003.
- [3] Triana User Guide - The Triana Project homepage, <http://www.trianacode.org/>
- [4] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Greenwood, T. Carver, M. R. Pocock, A.

- Wipat and P. Li. Taverna: a tool for the composition and enactment of bioinformatics workflows, *Bioinformatics*, Vol. 20 no. 17, 2004, pages 3045–3054.
- [5] P. Kacsuk and G. Sipos: Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal, *Journal of Grid Computing* Vol. 3. No. 3-4., 2005, Springer, 1570-7873, pp 221-238
- [6] The UK National Grid Service Website, <http://www.ngs.ac.uk/>
- [7] The EGEE web page, <http://public.eu-egee.org/>
- [8] The TeraGrid Website, <http://www.teragrid.org>
- [9] The Open Science Grid Website, <http://www.opensciencegrid.org/>
- [10] The P-GRADE portal Website, <http://www.lpds.sztaki.hu/pgportal/>
- [11] T. Delaittre, T. Kiss, A. Goyeneche, G. Terstyanzsky, S. Winter, P. Kacsuk: GEMLCA: Running Legacy Code Applications as Grid Services, *Journal of Grid Computing* Vol. 3. No. 1-2. June 2005, Springer Science + Business Media B.V., Formerly Kluwer Academic Publishers B.V. ISSN: 1570-7873, pp 75-90 JSR-000168 Portlet Specification, <http://jcp.org/aboutJava/communityprocess/review/jsr168/>
- [12] SRB project homepage - http://www.sdsc.edu/srb/index.php/Main_Page
- [13] P. Kacsuk, T. Kiss, G. Sipos, Solving the Grid Interoperability Problem by P-GRADE Portal at Workflow Level, Conf. Proc. of the Grid-Enabling Legacy Applications and Supporting End Users Workshop, within the framework of the 15th IEEE International Symposium on High Performance Distributed Computing , HPDC'15, Paris, France, pp 3-7, June 19-23, 2006
- [14] NGS JSDL Application Repository and Job Submission Portal, <https://portal.ngs.ac.uk/JobProfiles.jsf>
- [15] Xiaobo Yang, Dharmesh Chohan, Xiao Dong Wang, Rob Allan: A Web Portal for the National Grid Service, Conf. Proc. of the UK E-Science All Hands Meeting, 19 - 22 September 2005, Nottingham, UK, ISBN 1-904425-534 <http://www.allhands.org.uk/2005/proceedings/papers/445.pdf>
- [16] Biomedical Informatics Research Network Portal - <http://www.nbirn.net/cyberinfrastructure/portals.shtml>
- [17] Jason Novotny, Ramil Manansala, Thien Nguyen: BIRN Portal Overview, Portals & Portlets 2006, 17-18 July 2006, Edinburgh, UK <http://www.nesc.ac.uk/action/esi/download.cfm?index=3246>
- [18] The National Center for Microscopy and Imaging Research (NCMIR) – SRB portlet <http://ncmir.ucsd.edu/Software/srbportlet.htm>
- [19] myGrid project Website, <http://www.mygrid.org.uk/>
- [20] The Taverna Project Website, <http://taverna.sourceforge.net/>
- [21] Ali Anjomshoaa, et.al: Job Submission Description Language (JSDL) Specification, Version 1.0, <http://www.gridforum.org/documents/GFD.56.pdf>