Brokering Multi-grid Workflows in the P-GRADE Portal*

Attila Kertész[1,2], Gergely Sipos[2], and Péter Kacsuk[2]

[1] Institute of Informatics, University of Szeged
H-6720 Szeged, Arpad ter 2, Hungary
keratt@inf.u-szeged.hu

[2] MTA SZTAKI Computer and Automation Research Institute
H-1518 Budapest, P. O. Box 63, Hungary
CoreGRID Institute on Resource Management and Scheduling
{sipos,kacsuk}@sztaki.hu

Abstract. Grid computing has gone through some generations and as a result only a few widely used middleware architectures remain. The Globus Toolkit is the most widespread middleware in most of the current production grid systems, but the LCG-2 middleware dominates in Europe. The paper describes a brokering solution that enables the interoperability of various Globus and LCG-2 based grids during the execution of workflow applications, and supports users to utilize computing and storage resources from multiple production grids by a single application. The development and execution of such applications can be managed by a Web-based Grid portal called P-GRADE Portal, and the brokering of the workflows is carried out by its integrated GTbroker and LCG-2 broker component.

Keywords: Grid Computing, Grid Portal, Resource Broker, Workflow Management, Globus Toolkit.

1 Introduction

The Grid was originally proposed as a global computational infrastructure to solve grand-challenge, computational intensive problems that cannot be handled within reasonable time even with state of the art supercomputers and computer clusters [1]. Grid computing tackles these tasks by aggregating geographically and architecturally dispersed hardware and software resources into large virtual super-resources.

Meanwhile grids can be realized relatively easily by building a uniform middleware layer, such as Globus [2], on top of the hardware and software resources, the programming concept of such distributed systems is not obvious. Complex problems often require the integration of several existing sequential and parallel programs into a single application in which these codes are executed according

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

W. Lehner et al. (Eds.): Euro-Par 2006 Workshops, LNCS 4375, pp. 138–149, 2007.

to a graph, called workflow. The workflow concept introduces parallelism at two levels. The top level parallelism comes from the graph concept, i.e., codes contained by independent branches can be executed simultaneously. The bottom level parallelism can be applied if some of the workflow nodes are themselves parallel programs. Both top level and bottom level parallelism can be exploited if the parallel branches contain parallel nodes. In such case several supercomputers or clusters can be used simultaneously, and every parallel program would use one of these systems. Consequently, multi-site parallel application execution can be achieved without any performance degradation. The approach combines the benefits of traditional single-site parallel processing and grid-like multi-site processing. Although there are a large number of workflow-oriented grid activities, most of them do not exploit these two possible levels of parallelism [4][5][6].

After the proper parallel processing approach has been selected the next step is to choose a suitable application developer and execution environment. Grids are typically accessed through portals that serve as both grid application developer and executor environments. As grid technology matures the number of production grids dynamically increases. Although sometimes multiple grids are served by the same portal, usually different portals are installed for different grids. Even if a portal is connected to multiple grids, applications that utilize services from these grids simultaneously are not supported.

The P-GRADE Portal [20] is a workflow-oriented portal that supports applications that utilize services from multiple grids simultaneously and demonstrates how Web-based Grid portals can be implemented on top of the Globus middleware [2].

Executing a job in a grid environment requires special skills like how to find out the actual state of the grid, how to reach the resources, etc. Not only computer scientists, but also people from other scientific fields started to deal with this topic, because using the grid resources makes scientific development and research faster and produces better results. As the number of the users are growing and grid services are starting to become commercial, resource brokers are needed to free the users from the cumbersome work of job handling. Though most of the existing grid middlewares give the opportunity to choose the environment for the user's task to run, but originally they are lacking such a tool that automates the discovery and selection. Brokers meant to solve this problem. The Globus middleware does not provide brokering though it has an API that can be used to build such a tool. GTbroker [24] is a proper solution for this toolkit to provide automated job submission for the users.

This paper describes how this broker can be adopted by grid portals to reach Globus resources in an automated way. The following sections introduce the workflow management of the P-GRADE portal and the execution of the workflows with its incorporated brokers. This combination enables multi-grid brokering, even for different middlewares (Globus 2, 3, 4 and LCG-2). Since this portal is already connected to various grids, the automatic multi-grid workflow execution has become reality.

2 The P-GRADE Portal

The P-GRADE Portal is a workflow-oriented grid portal with the main goal to support all stages of grid workflow development and execution processes. It enables the graphical design of workflows created from various types of executable components (sequential, MPI [3] or PVM [17] jobs), executing these workflows in Globus-based [2] computational grids relying on user credentials, and finally, analyzing the monitored trace-data by the built-in visualization facilities. The P-GRADE Portal provides the following functions (see also Fig. 1.): Defining grid environments, creation and modification of workflow applications, managing grid certificates, controlling the execution of workflow applications on grid resources and monitoring and visualizing the progress of workflows and their component jobs.

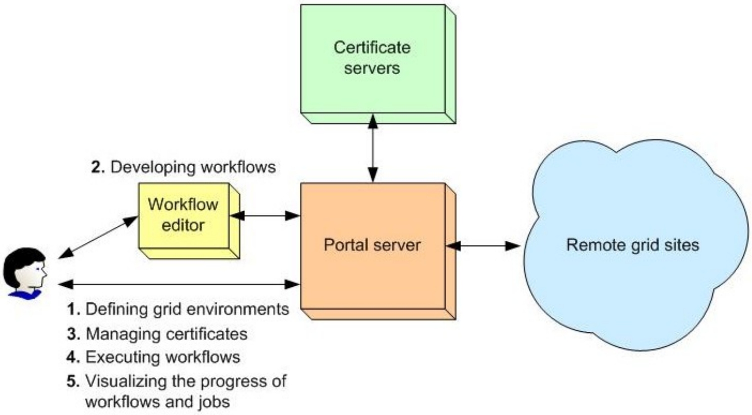


Fig. 1. User activities supported by the P-GRADE Portal

Current portals support only isolated users, i.e., grid users cannot collaborate via the portal either to develop applications together or collaboratively run existing applications. Multi-user portals provide controlled and concurrent access to grid applications for multiple users during both the application development and execution phases. This portal can connect several grids and able to support the simultaneous, collaborative execution of components of a workflow in several connected grids. The P-GRADE portal can give the users all these functionalities, so this portal is a collaborative-grid/user portal. In this paper we are focusing on workflow management. For more information on the portal please refer to [20].

3 Workflow Management in the P-GRADE Portal

Every workflow-oriented portal consists of a workflow GUI and a workflow manager part. While the workflow GUI is the interface that enables the development, submission and steering of workflows and the visualization of results, the workflow manager is responsible for the execution and scheduling of workflow components in the connected grids. It can simultaneously utilize multiple grids to execute different components of a workflow.

3.1 Workflow Notation

Workflow applications can be developed in the P-GRADE Portal by the graphical Workflow Editor. The Editor is implemented as a Java Web-Start application that can be installed on the client machines "on the fly", using a standard Web browser. The Editor communicates only with the Portal Server, and it is completely independent from the grid infrastructures the Server is connected to.

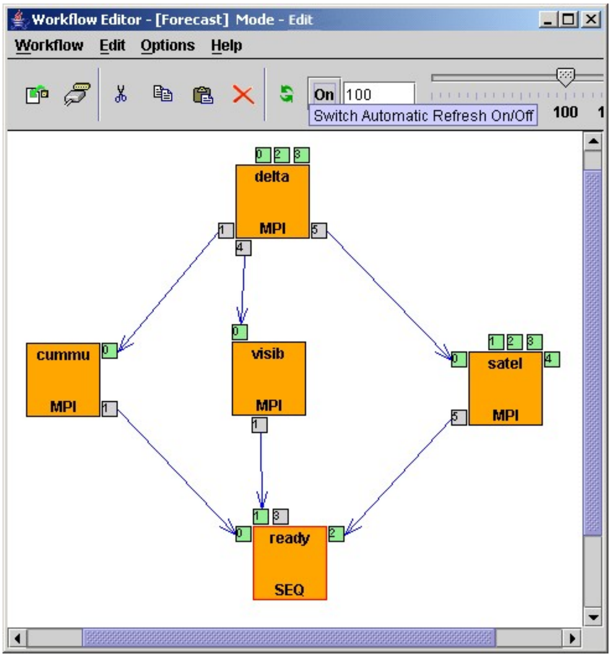
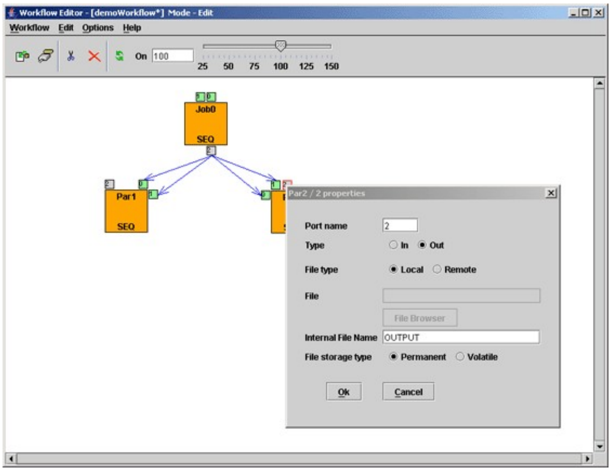


Fig. 2. A workflow graph in the P-GRADE Portal

A P-GRADE Portal workflow is a directed acyclic graph that connects sequential and parallel programs into an interoperating set of jobs. The nodes of such a graph are batch jobs, while the arc connections define data relations among these jobs. Arcs define the execution order of the jobs and the input/output dependencies that must be resolved by the workflow manager during execution (Fig. 2.).

Nodes labeled as delta, cummu, visib, satel and ready represent executable programs. Small squares labeled by numbers around the nodes are called ports and represent input and output data files that the corresponding executables expect or produce. (One port represents one input/output file.) Directed arcs interconnect pairs of input and output ports if an output file serves as an input file for another job. An input file – represented by an input port – can come from three different sources: It can be produced by another job of the workflow, come from the workflow developer's desktop machine or from a storage resource. An output file – represented by an output port – can also have the following three target locations: A computational resource, the Portal server or a storage resource (Fig. 3.).



On 100

Port name 2

Type ○ In ● Out

File type ● Local ○ Remote

File

Internal File Name OUTPUT

File storage type ● Permanent ○ Volatile

Fig. 3. Input/output file handling

The semantics of the workflow execution means that a node (job) of the workflow can be executed if, and only if all of its input files are available, i.e., all the jobs that produce input files for this job have successfully terminated, and all the other input files are available on the Portal Server and at the pre-defined storage resources. Therefore, the workflow describes both the control-flow and the data-flow of the application. If all the necessary input files are available for a job, then the workflow manager transfers these files - together with the binary executable - to the computational resource where the job is allocated for execution. Managing the transfer of files and recognition of the availability of the necessary files is the task of the workflow manager component of the Portal Server. In case of using the brokering service over Globus-based VO-s, GTbroker handles the necessary file transfers.

3.2 Developing and Editing Workflows

For simplicity let us examine a scenario, when a user works on workflows individually during both the development and execution phases. In the P-GRADE

Portal a workflow can be loaded from the user's private storage space – allocated on the Portal Server – into the client-side Editor, can be edited locally, and the updated version can be loaded back to the Server. The development of a P-GRADE Portal workflow consists of two subtasks: Defining the structure of the graph and specifying the properties of nodes (jobs and ports).

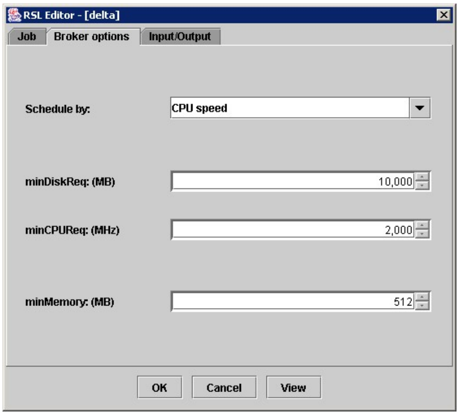
The graph structure can be defined using the drag and drop GUI elements of the Workflow Editor. The properties of nodes can be specified using property windows: by double clicking a job or a port a corresponding property window can be popped up and the attributes of the affected component can be defined.

The job component of a workflow node can be defined in the following way: using the job property window the user must specify the client side location and the type of the binary executable. Optional start-up parameters can also be given here (e.g. command line attributes). The job can be mapped onto a computational resource in the following way: Using the "Grid" and "Resource" dropdown listboxes first a grid, then a computational resource from that grid must be chosen. Jobs can be mapped onto resources of the VO the user has valid certificates to. If a broker is selected for job submission it can be seen from the "Grid" name (SZTAKI_MDS_2_BROKER – means that GTbroker is used for the job in the SZTAKI Grid). The portal can interface with 2 kinds of brokers: GTbroker for Globus 2 or 3 Grids and the broker component of the LHC Grid infrastructure [22].

The job requirements can be set in the Workflow Editor of the portal. GTbroker needs an RSL [2] file, which is created by the portal through the so-called RSL Editor. The other broker needs a JDL [22] file, created by the JDL Editor. They have a similar interface, so the user can use the same data to set the Job attributes. From the job property window the user can select the RSL Editor, when he/she has already chosen GTbroker for the "Grid" field. In this Editor window the redirection of standard streams and brokering options can be set, and a summary of the input/output files for the job can be viewed. The "Broker options" enables selection of resource mapping guidelines and defining minimum disk size, CPU speed and memory size requirements (Fig. 4.). Only this panel requires additional information about the job compared to the JDL Editor. The guidelines tell the broker to order the resources by disk size, CPU speed or memory size, or to use only clusters for execution environment. Clicking on "View" at the bottom of the window the generated RSL file can be viewed.

4 Workflow Execution

Because none of the largest production grids contain workflow manager services, workflow-oriented portals connected to them must incorporate workflow managers, too. The P-GRADE Portal contains a DAGMan-based [11] workflow manager subsystem which is responsible for the scheduling of workflow components in grids. This section discusses the workflow executor subsystems of the P-GRADE Portal with the brokering functions provided by GTbroker and the LCG-2 Broker.



Schedule by: CPU speed

minDiskReq: (MB) 10,000

minCPUReq: (MHz) 2,000

minMemory: (MB) 512

Fig. 4. The RSL Editor

4.1 Workflow Management in Details

One of the main goals of the P-GRADE Portal is to hide the low level details of grid systems with high-level, technology-neutral interfaces that can be easily integrated with different middleware. The GUI of the Portal is built with the GridSphere portal framework [19], thus the various portal functions are implemented as nearly independent portlets.

The "Certificate manager" portlet is responsible for uploading X.509 certificates into MyProxy servers [7] and for downloading short-term GSI proxies [12] into the workflow manager. These proxies are used for authentication. The "Settings" portlet can be used to specify Globus VOs and computational resources for the portal application. The "Workflow" portlet is the graphical interface of the workflow manager and can be used to submit and control workflows, to monitor and visualize execution.

The "Workflow" portlet is interfaced with the Condor DAGMan [11] workflow scheduler. DAGMan degrades workflows into elementary file transfer and job submission tasks and schedules the execution of these tasks. Although DAGMan itself cannot invoke grid services, it supports customized grid service invocations by its pre/post script concept [11]. One pre and one post script can be attached to every node of a DAGMan workflow. DAGMan guarantees, that it first executes the pre script, then the actual content script and finally the post script when it reaches a workflow node. Consequently, the Portal Server automatically generates appropriate pre, content and post scripts for every workflow node when the workflow is saved on the server. These scripts – started by DAGMan according to the graph structure –, invoke the GridFTP and GRAM clients to access files and start up jobs in the connected grids. DAGMan invokes these scripts in the same way in both single- and multi-grid configuration. In general, when a broker is used for job submission, the pre script invokes the broker, and the post script waits till the execution is finished. The broker provides information about the actual job status and the post script notifies the portal about the status changes.

4.2 Multi-grid Workflow Brokering

During workflow editing in the P-GRADE Portal the user has the opportunity to select a resource for each job to run on, or to let a broker choose one. Currently two brokers are used by the portal: the LCG-2 Broker and GTbroker (Fig. 5.).

Regarding Globus Grids, when the right order of the jobs is selected by DAGMan (according to the dependencies of the jobs), the actual job is given to GTbroker to find a suitable environment and guide the job through the submission process. This broker uses GT2 C API [2] functions to perform interaction with the Globus resources and job submission. For determining the available hosts in the grid it queries the MDS [2]. The job submission to resources is done through GRAM, and a GASS server [2] is used to put the files needed for the job to the remote host and to get back the result files if there are any. These tools enable this broker to work without additional software on Globus Grids. Since most of the current grids use this middleware, the simply adaptation makes this broker relevant.

When a job of the workflow is selected to run with GTbroker, the pre script executes the broker with an RSL file created by the RSL Editor. For QoS, user requirements are taken into account during resource selection. The extended RSL file contains the user requirements and job properties. Static and dynamic information are also used for matchmaking.

In Globus Grids the MDS contains the static properties of the appropriate VO resources. After getting the resources from the MDS, GTbroker orders them by a predefined criterion. In the criteria one can use the following metrics: CPU speed, number of CPUs, free CPUs on the node, disk size and whether a node is a cluster. With these metrics the hosts can be ordered in a way that the ones having the best resources for the actual job get higher priority than the others. The user can modify the priority by selecting the suitable one in the RSL Editor of the portal.

Dynamic information is also used by the broker. For PBS clusters the broker can determine the actual load, right before submitting the job to the selected resource. The pbsnodes command gives back the present availability and load of each node in the selected cluster. This additional piece of information makes the broker able to react for dynamic changes, and reject choosing an overloaded cluster. With this method it automatically finds the best resources and submits only jobs that can actually run.

Fault tolerance is supported by resubmissions. Should a job fail or be pending for too long on a resource (this time interval is set in the broker), the broker cancels and resubmits it to another high priority one. The actual state of the jobs is tracked by the broker, that's why it is possible to cancel and resubmit jobs. After the job is successfully finished, the result files are staged back by the broker and the workflow execution is continuing with the post script of DAGMan. The job states are sent to the portal, so it can visualize the execution phases of the job and therefore of the whole workflow. With this functionality the users are aware of the state of their workflows and notified about each step the execution is going through.

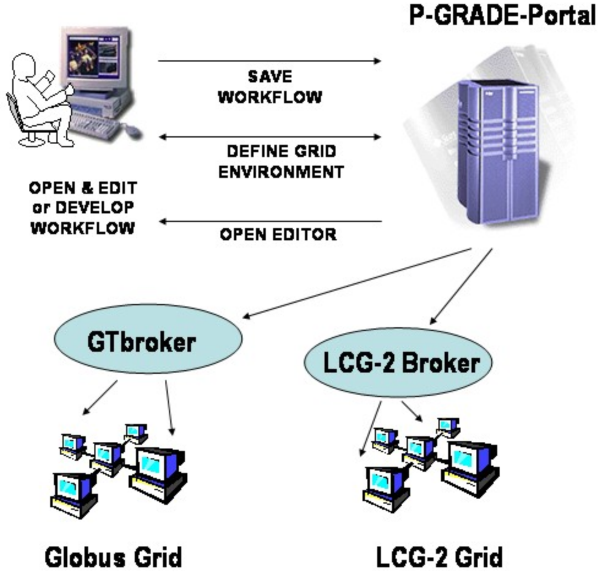


Fig. 5. Multi-grid workflow brokering

The LCG-2 brokering solution is also used by the P-GRADE portal to reach LCG-2-based grids. This kind of broker is built in the LCG-2 architecture: it uses the following parts: a User Interface machine is needed to use the Workload Management System. The Workload Manager is responsible for calling the Matchmaker, interacting with the Information System (BDII), the Replica Location Service, the Log Monitor and the Logging and Bookkeeping server. The Matchmaker gets the job data in a JDL file, and tries to find a "close" host: it takes into account the distance of the physical files on the Storage Elements to the actual Computing Element; finally it submits the job there. The WM can be informed of job failures through the Log Monitor, but automatic resubmission is not provided unlike in the case of GTbroker. Both solutions provide automatic workflow execution, but GTbroker relies only on Globus services and its usage is not limited to grids with LCG-2 architecture.

In the executable workflow the jobs are mapped to resources or brokers. The selected broker determines a VO for the job, from which the broker has to select the executing site. In the case of pure Globus-based grids GTbroker queries the MDS for resource information of the selected VO, and in LCG-2-based grids the LCG-2 broker asks the BDII. The nodes of the workflow that need LCG-2 services can be selected to run on a grid supporting them, and nodes that require only Globus services can be mapped to VOs handled by GTbroker (Fig. 5.). It means that a workflow can be brokered over several VOs, even on different grids. This multi-grid brokering is performed by the brokers connected to the portal. The portal is able to adopt other brokering services, therefore the number of reachable grids is growing.

5 Related Work

Workflows can be managed not only by grid portals, but by other traditional grid user interfaces and problem solving environments (PSE) as well. Unicore [13] and Triana [6] are two of the most well-known workflow-oriented PSEs. They provide neither multi-grid access, nor collaborative user support. Although the server of Triana is built on top of the GAT API [14] - thus it could abstract the underling grid services from their actual implementations - it cannot distinguish security domains from each other, which is a prerequisite of multi-grid access.

Pegasus [15] is a Web-based grid portal, which has the same isolated environment. Based on a special configuration file, filled up by the portal administrator with Globus GRAM and GridFTP [2] site addresses, Pegasus is able to map abstract workflows onto physical resources. At the same time – because of the centrally managed resource list and the single certificate the manager applies during workflow execution – Pegasus cannot be considered a multi-grid portal.

The GridFlow portal [16] applies a more complex workflow executor subsystem than the above discussed environments. The workflow manager of GridFlow handles workflows at two levels. It manages workflows at a global grid level and schedules them at the level of different local grids, but it does not provide collaborative development and execution capabilities.

KOALA [21] is a grid scheduler that uses some of the components of the Globus Toolkit, supports processor and data co-allocation and provides automatic resource selection. Users can interact with KOALA through so-called runners, which are command line tools and require RSL file specifications. KOALA has various runners for submitting and monitoring various kinds of jobs (MPI, Ibis). Currently it is only available on their own multicluster system (DAS 2 – Distributed ASCI Supercomputer 2), but they are planning to make it available on other grids.

Regarding brokers, several solutions have been developed up till now. These solutions usually require other tools to run and the user usually needs to modify its configuration or even additional software needs to be installed to the grid middleware. GTbroker is a broker for the Globus Toolkit, which uses only the APIs and services provided by the toolkit, performs automatic resource discovery and job submission with QoS and fault tolerant features. Since it does not need any other tools, it can be easily incorporated into portals.

6 Summary and Conclusions

With the most advanced portals multiple users can work together to define and execute grid applications that utilize resources of multiple grids. By connecting previously separated grids and previously isolated users together, these portals will revolutionize multidisciplinary research.

The P-GRADE Portal gives a Globus-based implementation for workflow management even for the collaborative-grid, collaborative-user concept [23]. Due to the multi-grid concept a single portal installation can serve user communities of multiple grids. These users can define workflows using the high-level graphical notations of the Workflow Editor, can manage certificates, workflows and jobs through the Web-based interface of the Portal Server. With exploiting the advanced workflow management features of the P-GRADE portal and the brokering functions of GTbroker and LCG-2 Broker users can develop and execute multi-grid workflows in a convenient environment. Users have access to more VOs can create such multi-grid workflows that reach resources from even different grids. Furthermore, the execution of these workflows is carried out in an efficient, brokered way. Since almost every production grid uses Globus middleware today, these grids could all be accessed by the P-GRADE Portal and the workflows created by the portal can produce the expected results.

P-GRADE Portal 2.1 [25] has been already connected to several European grids (LHC Grid [22], EU GridLab testbed [14], UK OGSA test-bed [8], UK NGS [10]) and serves as a graphical interface for several production grids like SEE-GRID [9], HunGrid [18] and UK NGS [10]. While the 2.2 version is already connected to the broker of the LCG middleware [22], the next, upcoming version is connected to GTbroker.

References

1. I. Foster, C. Kesselman, “Computational Grids, The Grid: Blueprint for a New Computing Infrastructure”, Morgan Kaufmann, 1998. pp. 15-52.
2. I. Foster C. Kesselman, “The Globus project: A status report”, in Proc. of the Heterogeneous Computing Workshop, IEEE Computer Society Press, 1998, pp. 4-18.
3. M. Snir, S. W. Otto, S. Huss-Lederman, D. W. Walker, J. Dongarra, “MPI: The Complete Reference”, MIT Press, 1995.
4. Ewa Deelman, et al, “Mapping Abstract Complex Workflows onto Grid Environments”, Journal of Grid Computing, Vol.1, no. 1, 2003, pp. 25-39.
5. Matthew Addis, et al: “Experiences with eScience workflow specification and enactment in bioinformatics”, in Proc. of UK e-Science All Hands Meeting (Editor: Simon J. Cox), 2003.
6. I. Taylor, et al., “Grid Enabling Applications Using Triana”, Workshop on Grid Applications and Programming Tools, Seattle, 2003.
7. J. Novotny, S. Tuecke, V. Welch, “An Online Credential Repository for the Grid: MyProxy”, in Proc. of 10th IEEE International. Symposium on High Performance Distributed Computing, 2001
8. UK e-Science OGSA Testbed: http://dsg.port.ac.uk/projects/ogsa-testbed/
9. Southern Eastern European GRid-enabled eInfrastructure Development (SEE-GRID): http://www.see-grid.org/
10. UK National Grid Service: http://www.ngs.ac.uk/
11. D. Thain, T. Tannenbaum, and M. Livny, “Distributed Computing in Practice: The Condor Experience”, Concurrency and Computation: Practice and Experience, 2005, pp. 323-356.
12. Butler, R., Engert, D., Foster, I., Kesselman, C., Tuecke, S., Volmer, J. and Welch, V. A National-Scale Authentication Infrastructure. IEEE Computer, 33 (12). 60-66. 2000.

13. D. W. Erwin and D. F. Snelling., “UNICORE: A Grid Computing Environment”, In Lecture Notes in Computer Science, volume 2150, Springer, 2001, pp. 825-834.
14. G. Allen et. al., “Enabling Applications on the Grid: A GridLab Overview”, International Journal of High Performance Computing Applications, Issue 17, 2003, pp. 449-466.
15. G. Singh et al, “The Pegasus Portal: Web Based Grid Computing” In Proc. of 20th Annual ACM Symposium on Applied Computing, Santa Fe, New Mexico, 2005.
16. J. Cao, S. A. Jarvis, S. Saini, and G. R. Nudd, “GridFlow: WorkFlow Management for Grid Computing”, In Proc. of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID’03), 2003, pp. 198-205.
17. V. Sunderam, J. Dongarra, “PVM: A framework for parallel distributed computing”, Concurrency: Practice and Experience, 2(4), 1990, pp. 315-339.
18. The HunGrid Virtual Organisation: http://www.lcg.kfki.hu/?hungrid&hungrid-general
19. J, Novotny, M. Russell, O. Wehrens: “Grid-Sphere: A Portal Framework for Building Collaborations” in Proc. of the 1st International Workshop on Middleware in Grid Computing, Rio de Janeiro, Brazil, 2003.
20. Csaba Németh, Gábor Dózsa, Róbert Lovas, Péter Kacsuk, “The P-GRADE Grid Portal”, Lecture Notes in Computer Science, Volume 3044, Jan 2004, pp. 10-19.
21. KOALA Co-Allocating Grid Scheduler: http://www.st.ewi.tudelft.nl/koala
22. LCG-2 User Guide, 4 August, 2005: https://edms.cern.ch/file/454439/2/LCG-2-UserGuide.html
23. Péter Kacsuk, Gergely Sipos, “Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal”, Journal of Grid Computing, Feb 2006, pp. 1-18.
24. A. Kertész, “Brokering solutions for Grid middlewares”, in Pre-proc. of 1st Doctoral Workshop on Mathematical and Engineering Methods in Computer Science, 2005.
25. P-GRADE Grid Portal: http://lpds.sztaki.hu/pgportal