

# The Grid Resource Broker, a ubiquitous grid computing framework

Giovanni Aloisio, Massimo Cafaro,  
Euro Blasi and Italo Epicoco  
*Department of Innovation Engineering, University of  
Lecce, Italy*  
*E-mail: {giovanni.aloisio, massimo.cafaro,  
euro.blasi, italo.epicoco}@unile.it*

Portals to computational/data grids provide the scientific community with a friendly environment in order to solve large-scale computational problems. The Grid Resource Broker (GRB) is a grid portal that allows trusted users to create and handle computational/data grids on the fly exploiting a simple and friendly web-based GUI. GRB provides location-transparent secure access to Globus services, automatic discovery of resources matching the user's criteria, selection and scheduling on behalf of the user. Moreover, users are not required to learn Globus and they do not need to write specialized code or to rewrite their existing legacy codes. We describe GRB architecture, its components and current GRB features addressing the main differences between our approach and related work in the area.

## 1. Introduction

Grid computing [1] is now regarded as one of the most promising approach to solve large-scale computational science problems. The key concepts of computational and data grids build on the foundations of high performance computing, distributed computing and high speed wide area networks. Grids can be thought of as a mixture of software and hardware infrastructure whose aim is to provide a coherent, unified view of geographically spread computing resources, smart instruments and distributed data archives. This seamless integration will enable users to share their resources and to build new classes of applications based on resource pooling and selection.

Current technologies to build computational grids include, among others:

- Legion [2], an object-oriented approach to grid computing;

- Globus [3], a layered services-oriented approach;
- Condor and Condor-G [4,5], a system for High Throughput Computing on the Grid;
- the Web, along with client-server architectures like Corba and Java RMI.

In this paper, we shall concentrate on the use of the Globus Toolkit and the Web to provide users with a secure, fast and reliable access to computational grids. It is worth noting here that computational grids and the Web are complementary technologies. We think of the Web as the perfect gateway for the client who wishes seamless access to services and data archives, obviating the need for special purpose, custom client software.

Besides fulfilling this role, the HTTP protocol increasingly provides a means for communication between machines as well between machines and people. Web servers are also hosting portals acting like high-level brokers, allowing location-transparent access to data archives, legacy systems and supercomputers. The main reasons for using the Web are ubiquity, portability, reliability and trust. While the Web is certainly not amenable to high performance computing (which is instead one of the most important uses of computational grids), it is a good choice to generate a request for supercomputer time or to start a distributed application.

The transition to grid computing is not easy. Researchers are not traditionally willing to spend their time learning new technologies and rewriting their legacy codes. This is especially true for people coming from fields other than computer science. The barrier of human-intensive software porting proves to be a large one, deterring many of those who would benefit.

Since the learning curve for Globus is particularly steep, we designed GRB [6] to promote the use of computational and data grids by providing an immediate access to Globus services, from a very general grid tool with a friendly web GUI. We do not require neither that users learn Globus, nor they write specialized code or rewrite legacy code. The interface hides all of the underlying Globus details, it is easy to understand and provides a number of functionalities. Thus, GRB

is a general framework for ubiquitous desktop access to grid resources. GRB builds on the Globus Toolkit, leveraging existing functionalities but providing a powerful interface to them. We utilize the following core Globus services to enable grid computing:

- Grid Resource Information Service (GRIS);
- Grid Index Information Service (GIIS);
- Globus Resource Allocation and Management (GRAM);
- Grid Security Infrastructure (GSI);
- Grid FTP (Grid-FTP).

The GRIS is a service running on each computing resource; it collects both static and dynamic resource information, and can also report the collected information to a hierarchical GIIS server posting and receiving data in LDIF format using the LDAP protocol (Lightweight Directory Access Protocol). Thus, the GIIS provides a uniform resource information service allowing distributed access to structure and data information about the grid status.

Resource management is the aim of the GRAM (Globus Resource Allocation Manager) module; it takes care of resource location and allocation; moreover it performs process management. The user can specify application requirements using RSL, the Resource Specification Language.

The Globus Security Infrastructure (GSI) module is responsible for handling the authentication mechanism used to validate the identity of the users and resources; it makes use of both the Generic Security Service API (GSS) and of SSL (Secure Socket Layer). Certificates are used as the principal authorization mechanism.

Remote access to data via sequential and parallel interfaces can be obtained exploiting the Grid-FTP server. The service is based on the GridFTP protocol and allows partial file, third-party and striped/parallel transfers, support for GSI authentication and negotiation of TCP buffer sizes.

The paper is organized as follows. Section 2 introduces the GRB; here we give a high level description of GRB components. A detailed description of current GRB functionalities is given in Section 3. We recall current state of the art in Section 4 addressing also the main differences between our approach and related work in the area. Finally, we conclude the paper in Section 5.

## 2. The GRB components

The GRB is a web gateway to the grid. Its main function is to mediate between the user's request and

the grid offering, hiding the complexity of the underlying middleware. The GRB grid portal is built on top of the GRB libraries [7,8] that in turn are layered on top of core Globus services. The libraries constitute a developer's toolkit, provide high level services and can be used to develop both web-based and desktop grid applications. In order to meet the user's requirements the GRB may need to access information services to retrieve useful information for automatic resource discovery and selection, or to stage on remote machines executables and/or input and output files. Once a pool of resources matching the user's criteria has been found, the GRB automatically starts the computation and it is responsible for monitoring the job execution. The job could be interactive or batch, may require multiple heterogeneous resources because its components are best executed on different architectures, or may require several homogeneous resources for parameter sweep studies. GRB also allows the execution of data flow tasks.

The GRB consists of several interacting components providing the required support:

1. *Resource Finder*: it is responsible for resource discovery, querying Globus information services like GRIS and GIIS using the LDAP API. On each computing resource where Globus is installed, there is an LDAP server called GRIS listening on port 2135. This server stores in LDAP entries the features of the machine. A GIIS server may or may not be available within an organization or institution and may be listening on an arbitrary port but, if available, it collects the information related to all of the computing resources from GRIS servers. Thus, it is a centralized directory service.
2. *Resource Matchmaker*: this module exploits the information returned by Resource Finder to find a subset of resources matching the user's criteria (like amount of memory, CPU type and speed, current load etc). If a suitable subset of computing resources has been found (the search may actually return an empty subset), the user can select interactively where to start the computation, driven by the information she gets back from the system or let Job Advisor schedule the job.
3. *File Assistant*: its purpose is to provide support for automatic staging on remote machines of executable and/or input/output files, using Grid-FTP servers that provide high performance file transfer (using the GridFTP protocol) coupled with enhanced user's authentication exploiting X.509v3

digital certificates and GSS API (Generic Security Service).

4. *Job Assistant*: this module provides support for all kind of job submissions. It works in conjunction with Resource Matchmaker, File Assistant, Parameter Sweep Agent, Data Flow Agent and Profile Manager. The user is asked to enter its delegated Globus credentials in order to authenticate her to the grid and, depending on the kind of submission, a minimal amount of information needed to specify completely the job. In this phase, the Profile Manager supplies the information previously entered, in order to minimize the time needed to actually start the computation. If needed, Grid-FTP servers are automatically contacted and files are transferred exploiting the services provided by File Assistant. This module interacts with Parameter sweep agent to allow submission of parameter study jobs, and with Data Flow Agent to allow submission of jobs described by a directed a-cyclic graph.
5. *Job Supervisor*: it is responsible for monitoring batch, parameter sweep and data flow executions, and coordinates itself with File Assistant to transfer output files once the job is completed.
6. *Parameter Sweep Agent*: this module provides support for submission of the same job with several inputs on multiple computing resources for parameter sweep studies. This functionality closely resembles the Condor system since it allows high throughput computing on the Grid. Here we exploit the Globus GRIS/GIS infrastructure for matchmaking (the term is borrowed from Condor) available resources against the user's criteria, while Condor utilizes its own infrastructure and a ClassAds mechanism. We do not provide support for checkpointing and migrating a job.
7. *Data Flow Agent*: it coordinates the execution of jobs described as a directed a-cyclic graph, taking care of precedence constraints.
8. *Profile Manager*: it handles user profiles, allowing a user to insert, update and remove information about grid resources. This information is used to speed up the job submission process, avoiding the need to re-enter it each time it is needed.
9. *Job Advisor*: this module provides support for scheduling jobs. Currently a round-robin strategy is used for scheduling parameter sweep jobs, while batch jobs are scheduled according to the required features of the machine, coupled with

the node/hour cost information found in the user's profile. Several scheduling algorithms such as economy driven scheduling [9] or scheduling in the presence of stale load information [10,11] will be developed.

### 3. GRB features

Our portal architecture is a standard three-tier model (Fig. 1). The first tier is a client browser that can securely communicate to a web server on the second tier over an HTTPS connection. The web server exploits Globus as grid middleware to provide its clients with a number of grid services on the third tier, the computational grid. The Globus toolkit provides the mechanisms for submitting jobs to remote resources apart from resource schedulers, querying for static and dynamic information about resources composing the computational grid using the LDAP API, and a secure PKI infrastructure that uses X.509v3 digital certificates.

There are no restrictions about what systems/sites could be served by GRB, because of the way user profiles are handled. As a matter of fact, GRB can be accessed regardless of system/geographical location and grid resources can be added/removed dynamically. In order to use GRB, a user must apply to the ISUFI/HPC Lab (University of Lecce) to get an account, to the Globus Certification Authority to get a certificate, and she must properly set up her Globus proxy on the machine running the MyProxy [12] server. By taking advantage of the MyProxy package, users can gain access to remote resources through the portal without requiring their certificate and private key to be located on the GRB machine running the web browser. The MyProxy server is responsible for maintaining user's delegated credentials, proxies that can be securely retrieved by a portal for later use.

We assume that Globus is installed on each one of the computing resources that the user adds to her profile; moreover we assume that Grid-FTP is installed and listens for incoming connections on the IANA registered port 2811, that the GRIS server is listening on the IANA registered port 2135 and that it can be queried starting from the distinguished name "Mds-VO-name = local, o = Grid". The user's client browser must be configured to accept cookies. The cookies are used for session management and user's authentication, they are encrypted using SSL and sent to the client's browser.

To start using the GRB, a user authenticates her to the system by means of her login name and the PEM

pass phrase (Privacy Enhanced Mail) that protects her Globus proxy stored on the MyProxy server. The transaction exploits the HTTPS (SSL on top of HTTP) protocol to avoid transmitting the user's PEM pass phrase in clear over an insecure channel. Once authenticated, a user can start a GRB session (Fig. 2) that will last for a user's specified number of hours (as decided at proxy creation time) or until she invalidates her session using the logout tool.

We now give a brief description of current GRB functionalities. In-depth technical details about the underlying implementation, including design issues, can be found in [7,8].

### 3.1. Adding New Resources to the user's profile

Before using the GRB, the user must create her user's profile. This file contains information about the computational resources on the Grid that the user can access. For each machine the user is asked to enter the hostname (required to contact Globus servers), the pathname to her favourite shell (required to access the user's environment on the remote machine), and a float number representing the node cost per hour in her currency (to be used for job scheduling). We plan to add more information in the future to allow for more complex scheduling algorithms. This functionality allows the users to quickly set up their user profiles and is provided by Profile Manager.

### 3.2. Modifying or deleting a resource from the user's profile

This tool allows the users to edit resource information in order to modify the shell pathname and/or the node/hour cost information. If the resource is no longer accessible, it should be deleted. Profile Manager provides the functionality.

### 3.3. Viewing the user's profile information

Users can quickly browse the information related to their computational resources by using this tool. The hostnames listed here represent the user's current grid. Profile Manager provides the functionality.

### 3.4. Given a specific computational resource, searching its GRIS to determine its features

Computational resources where Globus is installed can be queried about their features. The GRIS, populated by several Globus information providers, is a small LDAP server that stores both static and dynamic information about hardware and system software. Thus the GRIS can be thought of as a white pages service, and allows easy retrieval of the features of a machine. GRB interacts with GRIS servers using its Resource Finder module and the LDAP API.

### 3.5. Searching a GIIS for computational resources with given features

An institution or organization may decide to set up a GIIS service on one of its computational resources. Like the GRIS, the GIIS is an LDAP server, which collects the information, related to each GRIS server available in the institution or organization. Thus the GIIS server can be exploited like a yellow pages service, and allows users to find, if available, computing resources with specific features like amount of memory, number of processors etc. GRB interacts with GIIS servers using its Resource Finder module and the LDAP API.

### 3.6. Submitting a batch job

Once a machine has been found, maybe by a query to a GRIS or GIIS server, a user can submit a batch job supplying to Job Assistant just the executable name (no absolute pathname is needed here, because we exploit the user's environment on the remote machine), optional command line arguments and a path to the input and/or output file(s) (if needed). The executable and the input/output file(s) do not need to be stored on the machine chosen for execution. Indeed, we allow the use of several Grid-FTP servers to stage the executable and the input/output file(s) to/from the user's selected machines using the services provided by File Assistant. Profile Manager automatically supplies the hostnames required to identify the machines to be used for the computation, including Grid-FTP server hosts.

If job submission is successful, the user can see on the browser a job identifier to be used for checking the job status using Job Advisor. This information is saved automatically on the GRB web server machine, and it is used later when the user wants to check the job status. We recall here the possibility to start even a graphical client. Job Assistant automatically sets the Unix envi-

ronment variable DISPLAY to the client's IP address, so that the user needs just to authorize the grid resource to redirect the display on her machine using the "xhost" command on Unix machines or configuring as needed her X-server prior to job submission. Therefore, GRB users are not restricted to text-based applications, they can also start and steer graphical applications. Moreover, this implies the possibility to edit, compile and debug code by simply starting an "xterm" client.

### 3.7. Submitting an interactive job

In case of an interactive job submission, Job Assistant coordinates itself with Profile Manager to help the user enter the information required and with File Assistant to stage both the executable and the input file(s). The output is sent directly to the client browser. We recall here that "interactive" in this context does not mean that the users are allowed to enter data in response to an input request. If needed, this kind of interaction is allowed by GRB starting in batch mode an "xterm" graphical client and running the application inside.

### 3.8. Search a GIIS for machines with specific features and submit a job

This tool is a generic resource broker: it allows the user to search a GIIS server to find required computational resources (using Resource Finder and Resource Matchmaker) and to submit a batch job on one of the resources matching the specified criteria (using Job Assistant, Profile Manager and File Assistant). Scheduling is automatically performed by GRB, also taking into account the node/hour cost information stored in the user's profile.

### 3.9. Checking a batch job status

After submitting a batch job, there is the need to monitor the progress of the job. This tool allows the users to check a job status exploiting Job Supervisor and the information previously saved at job submission time. If PENDING, the job is sitting in a queue, waiting to be executed; during execution its status is ACTIVE. The job can alternate its status between ACTIVE and SUSPENDED, maybe because of pre-emption mechanisms. If something goes wrong, the job enters the FAILED state, otherwise normal termination is signalled by the DONE status. When the job has reached the DONE status GRB automatically transfers the output file(s) to a user's selected machine using Grid-FTP, if this service was requested.

### 3.10. Submitting a parameter sweep job

A number of users need to run the same executable with different input, e.g. to do a parameter study. This tool provides a convenient means to do this. We assume here that the user's input files are called input-1, input-2, ..., input-n. Moreover, we assume that all of the input files are stored on the same machine and allow output files to be transferred on a machine possibly different from the machines hosting the executable and input files. Grid-FTP is used if needed for automatic staging of executable and input/output files.

### 3.11. Checking a parameter sweep job status

This tool allows the users to check a parameter sweep job status. For each sub-job whose status is DONE, GRB automatically transfers the associated output files too, using Grid-FTP, if the user previously requested this service.

### 3.12. File Transfer

This tool allows the user to transfer files between machines using Grid-FTP. GRB supports single file transfer and directory transfer. The interface exploits DHTML to show the contents of the remote file system in terms of files and folder, and to allow for dynamic navigation inside the file systems (Fig. 3). Files are transferred using Grid-FTP third-party mechanism.

### 3.13. Data Flow tasks

GRB provides the users with an applet that allows deriving visually a data flow task (Fig. 4). The task is made of a number of sub-jobs whose precedence constraints are modelled using a directed a-cyclic graph. Vertices represent batch jobs and edges represent precedence within sub-jobs.

### 3.14. Invalidating the user's session

If the users do not need the services provided by the GRB any more, they should logout from the system invalidating their GRB web session. Anyway, their session will expire automatically after a user's specified number of hours.

#### 4. Related work

Other projects that recognize the importance of merging together computational grids and the Web are:

- WebFlow [13];
- UNICORE [14,15], Uniform Interface to Computing resources;
- SDSC GridPort Toolkit [16];
- NLANR Grid Portal Development Kit [17].

WebFlow uses a network of Java Servers, Corba and an applet front-end to support visual component based programming. WebFlow can be thought of a Web version of AVS or Khoros and integrates program modules together using a dataflow paradigm. The module developer defines data input and output interfaces and builds methods to handle data I/O. WebFlow has been used as a distributed middle tier to build Gateway, a commodity-based web portal that allows secure access to unclassified Department of Defense computational resources. The Gateway portal can handle job submission, file transfer and job monitoring, however these features were developed independently of typical grid services such as those provided by Globus, Legion and Condor.

The goal of UNICORE is to deliver software that allows users to submit jobs to remote high performance computing resources without having to learn details of the target operating system, data storage conventions and techniques, or administrative policies and procedures at the target site. Existing Web-based technologies are exploited wherever possible. The user interface is based on Java and modern browser technology to allow access to UNICORE resources from anywhere in the Internet for properly authorized users and to eliminate software distribution. A Network Job Supervisor (NJS) at each UNICORE site interprets the Abstract Job Object (AJO) generated by the user interface, manages the jobs and the necessary data. NJS interoperates with vendor specific batch systems, e.g. Cray NQE, IBM Load Leveler, Codine, etc. UNICORE focuses on the typical supercomputing mode of working, i.e., batch processing. None of the services provided relies on standard grid middleware such as Globus or Legion.

The Grid Portal Toolkit (GridPort) is designed to aid in the development of science portals on computational grids like user portals, applications interfaces, and education portals. GridPort provides information services that portals can access and incorporate, leveraging standard, portable technologies. GridPort exploits advanced web, security and metacomputing technologies

such as PKI and Globus to provide secure, interactive services.

HTML pages are built from server-side Perl/CGI scripts, and simple HTML/JavaScript on the client side. Portals built on top of GridPort allow users to execute jobs and to manipulate data and files through a web interface. Examples of portals built using GridPort are HotPage [18], LAPK [19], NBCR Heart [20] and GAMESS [21].

The Grid Portal Development Kit provides access to Grid services by using Java Server Pages (JSP) and Java Beans. Java Server Pages invoke Bean methods to authenticate users, manage profiles, submit jobs, etc. The GPDK Java beans are built on top of the Globus Java Commodity Grid (CoG) toolkit [22].

GPDK Java beans present an easier interface for web developers to use the CoG kit when developing portal server pages.

Since both WebFlow and UNICORE are not Globus based, we believe that it would be unfair to compare them with GRB. Nonetheless, to the best of our knowledge, there are the following differences. WebFlow can access information resources just to inform the user about the codes and machines available. There is not the possibility of interactive, data flow and parameter sweep job submission. For batch jobs, brokering on behalf of the user is not provided, and automatic staging of executable/input/output files to/from remote machines is missing. Remote execution of X-windows based graphical clients is not allowed either. File transfer utilizes standard FTP protocol, and cannot handle a directory tree; transfers include only the files in the chosen directory, not any subdirectories.

UNICORE does not access any information servers. There is not the possibility of parameter sweep job submission or brokering on behalf of the user, however meta-computing is allowed through advance reservation, a prerequisite for co-scheduling. Automatic staging of executables to/from remote machines is missing. Remote execution of X-windows based graphical clients is not allowed either. File transfer utilizes standard FTP protocol.

GridPort does not allow parameter sweep and data flow job submission. Automatic staging of executable/input/output to/from remote machines is missing. Remote execution of X-windows based graphical clients is not provided either. File transfer utilizes the GASS protocol and does not handle a directory tree, although, being integrated with the Storage Resource Broker (SRB) [23], GridPort can transfer a directory if the data collection belongs to SRB. Brokering is not allowed.

GSDK shares with GRB lots of features. The main differences lie in job submission capabilities. GSDK does not provide parameter sweep and data flow tools, and does not allow staging the executable. Brokering is not offered, not even remote execution of X-windows based graphical clients.

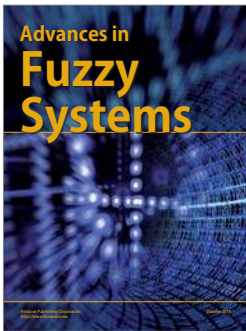
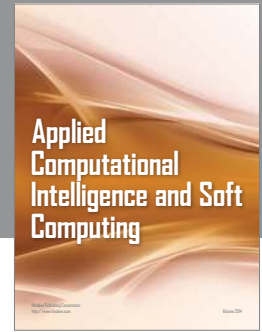
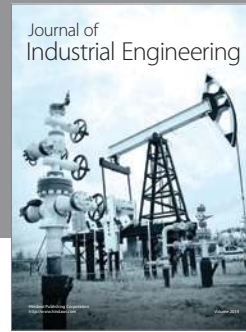
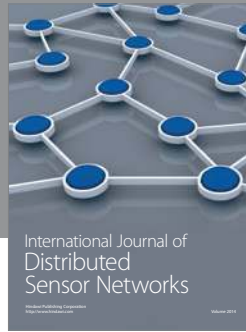
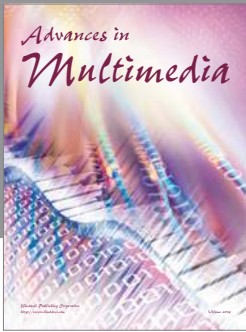
Main differences among GRB, GridPort and GSDK are reported in Table 1.

## 5. Conclusions and future work

A grid portal, the Grid Resource Broker, was presented. We showed its components and three-tier architectural model, and discussed its current features, highlighting GRB usefulness for users who do not want to learn how to use Globus for their grid computing needs. GRB was designed to be a very general grid tool providing location-transparent and secure access to Globus services using a simple web-based GUI. Automatic discovery of grid resources matching the user's criteria, and scheduling on behalf of the user are also provided. There are no restrictions to what systems/sites could be served by GRB, because of the way user profiles are handled. As a matter of fact, GRB can be accessed regardless of system/geographical location and Grid Resources can be added/removed dynamically. Moreover, users do not need to write specialized code or to rewrite their existing legacy codes. We plan to add other features, and to enhance the basic functionalities provided, implementing several scheduling algorithms that will also take into account the information provided by the Network Weather Service (NWS) [24]. GRB libraries are also exploited to build domain specific portals like remote sensing and medical imaging. A demonstration of the GRB capabilities was given in the SuperComputing 2000 conference, held in Dallas, Texas. Another demonstration was given in the context of the NPACI All Hands Meeting 2001, San Diego, California.

## References

- [1] I. Foster and C. Kesselman, eds, *The Grid: Blueprint for a new Computing Infrastructure*, Morgan Kaufmann Publishers, 1998, ISBN 1-55860-475-8.
- [2] A.S. Grimshaw, W.A. Wulf, J.C. French, A.C. Weaver and P.F. Reynolds Jr., The Legion Vision of a Worldwide Virtual Computer, *CACM* **40** (1997).
- [3] I. Foster and C. Kesselman, GLOBUS: a Metacomputing Infrastructure Toolkit, *Int. J. Supercomputing Applications* (1997), 115–128.
- [4] J. Basney, M. Livny and T. Tannenbaum, High Throughput Computing with Condor, *HPCU news* **1**(2) (June 1997).
- [5] J. Frey, T. Tannenbaum, I. Foster, M. Livny and S. Tuecke, Condor-G: A Computation Management Agent for Multi-Institutional Grids, *Proceedings of the Tenth IEEE Symposium on High Performance Distributed Computing*, (HPDC10), 2001.
- [6] G. Aloisio, M. Cafaro and R. Williams, The Grid resource Broker, <http://sara.unile.it/grb/grb.html>.
- [7] G. Aloisio, M. Cafaro, E. Blasi, L. Depaolis and I. Epicoco, The GRB Library: Grid Programming with Globus in C. Proceedings of HPCN Europe 2001, Amsterdam, Netherlands, Lecture Notes in Computer Science, Springer-Verlag, 2001, pp. 2110: 133–140.
- [8] G. Aloisio, M. Cafaro and I. Epicoco, Early experiences with the GridFTP protocol using the GRB GSI-FTP library, submitted to FGCS Journal special issue on Grid Computing.
- [9] R. Buyya, J. Giddy and D. Abramson, A Case for Economy Grid Architecture for Service-Oriented Grid Computing, 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), In conjunction with IPDPS 2001, San Francisco, California, USA, April 2001.
- [10] M. Mitzenmaker, How useful is old information, *IEEE Trans. Parallel and Distributed Systems* **11**(1) (Jan 2000).
- [11] M. Dahlin, Interpreting stale load information, *IEEE Trans. Parallel and Distributed Systems* **11**(10) (Oct 2000).
- [12] J. Novotny, S. Tuecke and V. Welch, An Online Credential Repository for the Grid: MyProxy, Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, 2001.
- [13] D. Bhatia, V. Burzevski, M. Camuseva, G. Fox, W. Furmanski and G. Premchandran, WebFlow – A Visual Programming Paradigm for Web/Java based coarse grain distributed computing, *Concurrency Practice and Experience* **9** (1997), 555–578.
- [14] M. Romberg, The UNICORE Architecture: Seamless Access to Distributed Resources, Proceedings of the eighth IEEE International Symposium on 'High Performance Distributed Computing' HPDC-8, IEEE Computer Society, Los Alamitos, CA, August 3–6, 1999, pp. 287–293.
- [15] J. Almond and D. Snelling, UNICORE: uniform access to supercomputing as an element of electronic commerce, *FGCS* **15**(5–6) (October 1999), 539–548.
- [16] SDSC GridPort Toolkit, <https://gridport.npaci.edu>.
- [17] NLANR Grid Portal Development Kit, <http://dast.nlanr.net/Features/GridPortal>.
- [18] M. Thomas, S. Mock and J. Boisseau, Development of Web Toolkits for Computational Science Portals: The NPACI Hot-Page, Proceedings of High Performance Distributed Computing, 2000.
- [19] R. Jelliffe, LAPK, <https://gridport.npaci.edu/LAPK>.
- [20] NBCR Heart, <https://gridport.npaci.edu/Heart>.
- [21] K. Baldrige and J. Greenberg, GAMESS, <https://gridport.npaci.edu/GAMESS>.
- [22] G. von Laszewski, I. Foster, J. Gawor, W. Smith and S. Tuecke, CoG Kits: A Bridge between Commodity Distributed Computing and High-Performance Grids, accepted to the ACM 2000 Java Grande Conference, 2000.
- [23] C. Baru, R. Moore, A. Rajasekar and M. Wan, The SDSC Storage Resource Broker, Proc. CASCON'98 Conference, Toronto, Canada, Nov. 30–Dec. 3, 1998.
- [24] R. Wolski, N. Spring and J. Hayes, The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing, *Future Generation Computer Systems* **15**(5–6) (1999), 757–768.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

