Chapter 20

THE GRID: AN ENABLING INFRASTRUCTURE FOR FUTURE E-BUSINESS, E-COMMERCE AND E-GOVERNMENT APPLICATIONS

Fabrício Silva, Hermes Senger

Abstract: In this paper we discuss the utilization of grid computing platforms as an enabling infrastructure for e-commerce, e-government and e-business applications. First fundamental concepts related with grid computing are presented, as well as a description of the evolution of grid computing platforms from pioneer projects to third generation systems. Then we identify a set of services that can be provided by grid computing platforms which will be fundamental for future e-commerce, e-business and e-government applications, such as database access and integration and knowledge discovery services. We also discuss why the grid may be the platform of choice for providing such services in a geographically distributed area.

Key words: Grid computing, e-business, e-commerce, e-government

1. FUNDAMENTAL CONCEPTS

A Grid is a system that coordinates heterogeneous, geographically distributed resources that are not subject to centralized control, using standard, open, general-purpose protocols and interfaces to deliver non-trivial qualities of service [17]. Grid platforms enable the sharing, selection and aggregation of a variety of resources including supercomputers, servers, workstations, storage systems, data sources and specialized devices that are geographically distributed and owned by different organizations [18]. Grids became possible in the last few years due to the great improvement in performance and the reduction in cost of both computer networks and microprocessors

The grid is characterized by four main aspects [2]:

- *Multiple administrative domains and autonomy.* Grid resources are geographically *distributed* across multiple administrative domains and owned by different organizations.
- *Heterogeneity. A grid involves a multiplicity of resources that are heterogeneous in nature and will encompass a vast range of technologies.*
- *Scalability. A grid may grow from few integrated resources to millions. This raises the problem of potential performance degradation as the size of the grid increases.*
- *Dynamicity or adaptability. In a Grid, resource failure is the rule rather that the exception. Resource managers and applications should adapt themselves in order to use the available resources efficiently.*

There has been considerable effort over the past few years in the development and implementation of platforms for grid computing, such as Condor [30], Legion[26] and Globus[25], to name a few. Condor supports the execution of applications making use of a single machine, pools of machines or machines already in a grid. Condor leverages the exploitation of idle cycles to provide a number of relevant services, such as task checkpoint and task migration. Legion implements an object model for grid computing. Within Legion, everything is represented as objects, from users, programs, storage, processors to special devices. Globus implements a set of infrastructure services to support the execution of applications on computational grids. The collection of services includes discovery and allocation of resources, security, information services, file transfer, and others. The Globus project has strongly influenced the evolution of research on grid computing, by proposing a set of standard mechanisms for creating, managing and exchanging information among service entities.

Originally the Grid has focused on the integration of computing resources, but as more of the possibilities for grid solutions have been realized this resource view has become complex. The same computing resource needs to be accessed and viewed very differently depending upon the application. It becomes much easier viewing the Grid as a collection of services and abstracting away from the underlying resources [31]. Beyond that, one can view the grid as a platform capable of integrating resources in different levels, from computing resources to metadata and knowledge resources.

One of the main reason for that change was the recent proposition of the Open Grid Services Architecture (OGSA) [22], and the associated Open Grid Service Infrastructure specification [38]. The Open Grid Services Architecture (OGSA) has been proposed as an enabling infrastructure for systems and applications that require the integration and management of services within distributed, heterogeneous, dynamic "virtual organizations"[18]—whether within industry, e-science, or e-business. The Open grid service architecture define conventions and WSDL interfaces related to grid services, a potentially transient stateful service instance supporting reliable and secure invocation, lifetime management, notification, policy management, credential management and visualization.

In the following section we describe in more detail the evolution of grid computing, from the first pioneer projects to the definition of the OGSA specification. Section 3 discusses the application of grid computing in e-business, e-commerce and e-government applications by describing services that can be made available through a grid platform. We believe that these services will play an important role in the infra-structure necessary for delivering the quality of service needed for e-commerce, e-business and e-government applications in the 21st century.

2. THE EVOLUTION OF THE GRID

In this section we provide a short overview of the evolution of grid computing, from its beginning around the late 1980s, until nowadays, focusing on the current perspective. Essentially, grid computing emerged from the need to share resources among individuals, and institutions, in a secure, flexible and coordinated way [2]. Although these main ideas are somehow present in the most of grid projects, there are some differences between what grid computing is meant to be in the past, and what are the main ideas which are currently driving its evolution. De Roure et. al. [12] depicts an evolution scenario in which grid projects can be classified into three generations, which we describe below.

2.1 The First Generation

From the beginning until the middle of 1990s, grid computing initiatives were driven to support a specific group of high performance applications with computational power. This is the first generation, and its major representatives include the FAFNER [15] and the I-WAY [19] projects. FAFNER was created as though a consortium led by Bellcore Labs., Syracuse University and Co-Operating Systems, and created a hierarchical network of web servers to coordinate a distributed effort for factoring big numbers. The main objective was testing the limits of security of the RSA encryption algorithm. The I-WAY project was started in 1995, to implement a high performance wide area backbone to connect seventeen U.S. high performance computing centers. In this project, attempts were made to provide a standardized solution for issues concerning security, scalability and heterogeneity, supporting the execution of large and complex applications. Both of these projects were led to provide computational resources to a range of high performance applications and influenced many other initiatives.

2.2 The Second Generation

The following years have witnessed a plethora of projects and initiatives launched to confront the challenges which feature grid computing systems, including those mentioned in section 1. Because grid applications can make use of a number of multi-institutional resources potentially ranging from dozens to thousands or millions, heterogeneity may become a big challenge. To cope with this, the middleware has emerged as a solution by implementing programming models which provide transparency from architecture details, such as network, operating system, computer hardware, programming languages and physical location. To enable scaling up to thousands or millions of geographically dispersed resources without degrading performance, applications were required to deal with higher latencies, and explore locality of resources they use. Also, because scalability involves traversing multiple organizational boundaries, much effort was issued to support authentication and trust issues. Furthermore, to cope with frequent resource failures, applications had to tailor their behavior to function without losing their functionalities and with maximum performance. Globus [21] and Legion [26] are representatives of second generation middleware for grid computing.

Globus [21] has been developed under an U.S. multi institutional project, to provide a software infrastructure that enables applications to handle distributed computing resources as a single virtual machine. Globus supports a set of basic infrastructure services with well defined APIs. Security, file transfer, resource location and resource management are examples of services implemented in Globus. It implements a multi-layered architecture, in which applications may be composed by aggregating lower level services provided by the Globus infrastructure, by specific tools, or by other applications. Most of these services have been realized in Globus Toolkit 2 (GT2), which is one of the most deployed software for building computational grids. Currently, Globus Toolkit 3 (GT3) provides a reference implementation for OGSA (see section 1) architecture and services, a standard created by the Global Grid Forum, which defines a set of Grid Services as specializations of Web Services to implement the grid.

Legion [26] is an entirely object-oriented system developed at the University of Virginia, which provides an infrastructure for building computational grids. Issues concerning scalability, heterogeneity, security, and resource management are addressed to provide a software infrastructure that enables the execution of applications that handle a great number of resources geographically distributed. Legion maintains the local administrative autonomy over the resource usage by letting local administrators to specify resource sharing policies. Legion supports a unique and persistent namespace for publishing resources and information to the applications. In 1998, the Applied Metacomputing Corp. was created to exploit Legion commercially and in 2001 it was relaunched as Avaki [1].

Furthermore, many other initiatives have been delivered to address specific issues or particular approaches related to grid computing. Some of them focus on handling from dozens to millions of multi-institutional resources. An example is Condor [13], a project started in 1985 at Univ. of Wisconsin at Madison as though a scheduler for idle resources for locally distributed computing system, which evolved to a grid solution to execute high throughput applications. Condor major features include detection and exploitation of computational resources, job management, and reliable executions through checkpointing, migration and restarting of failed jobs. Many other systems provide capabilities for resource management, and support the specification of resource scheduling policies, resource sharing policies, job requirements, execution monitoring, and fault tolerance. Examples include the Load Sharing Facility (LSF) [46] by Platform Computing Corp. [36], the Sun Grid Engine (SGE) [41], and the Portable Batch System (PBS) [42]. In addition, storage can be viewed as a resource. The Storage Resource Broker (SRB) [37] supports homogeneous access to a wide range of storage devices.

Peer-to-peer computing [8] provides a very plausible treat for scalability and fault tolerance issues through decentralization. With peer-to-peer, computers can share resources such as spare computing cycles, storage capacity, and databases, through the network without incurring in bottlenecks as it happens with the traditional client/server model. Some examples of P2P based distributed storage systems in the grid context include the FARSITE, the OceanStore, the Self-Certifying File System (SFS), and the PAST [12]. Also, web technologies have been used to build grid portals, which enable scientists and researchers of specific interest communities to access grid resources. A grid portal may provide user authentication, resource scheduling facilities, and access to remote information. Examples of grid portals include the NPACI HotPage (https://hotpage.npaci.edu/), the SDSC Grid Port Toolkit (http://gridport.npaci.edu/), and the Grid Portal Development Kit (http ://doesciencegrid.org /projects/GPDK/).

2.3 The Third Generation

Although problems which are inherent to geographically distributed systems (e.g. scalability, heterogeneity, security and adaptability) are still motivating many research activities, a new vision of grid computing is emerging, which goes beyond viewing the grid in terms of its enabling technologies. Instead, a more holistic view of the grid is emerging as other evolution aspects became apparent. This new envisioning of the grid is much more concerned with requirements, i.e., the features needed to support known applications such as e-Science, and what features can enable new applications which were not initially targeted to run on the grid [18], such as e-Business and e-Government.

As more applications have been implemented and deployed, the need for reuse application and information components became evident. Also, service orientation and use of metadata are increasingly present in grid systems. By combining these two features, grid resources can be assembled in a flexible manner to compose grid applications. Metadata are needed to cope with the automated discovery of functionality and availability of a great number of heterogeneous and geographically distributed resources. It is important as grid scales beyond the human capacity to manage and assemble resources manually. In addition, service orientation allows independence from the programming model, because the only public information about services are their interfaces and public metadata. Together, these features leverage the opportunities for system integration.

The service oriented vision of the grid approximates to other related research communities, such as the Web Services and Service Oriented Architectures. The World Wide Web Consortium (W3C) are leading the specification and deployment of standards (e.g. SOAP, WSDL, UDDI and RDF) to support a service oriented approach. Other important developments provide support for the process level. The IBM's Web Services Flow Language (WSFL) [43] allow workflows to be viewed as services, and Microsoft's XLANG [44] supports transactions that involve multiple web services. There are also initiatives to cope with the design of web services systems, such as the Web Services Modeling Framework [45]. As one can notice, there is a common way for the evolution of web services and grid systems. The emergence of the OGSA architecture is an evidence of the converging interests of grid computing and web services communities. The OGSA specification is an initiative launched by IBM and the Globus Project, which are currently being led in the context of Global Grid Forum (GGF) [24], a community that aggregates over four hundred organizations and five thousands individuals in order to promote and support the development of grid technologies and applications. At the mainstream of this evolution is the concept of Grid Services, which are grid enabled specializations of Web Services. Grid Services are entities independently created and deployed by Virtual Organizations [18]. Thus, the specification and deployment of standards is a keystone which can leverage both interoperability and competition among applications, toolkits and base services, implemented by different organizations. Interoperability allows building grid systems by mixing, matching, and aggregating different and competing implementations. Such degree of integration can be though as a major characteristic of third generation, since it was not achieved during the second generation of grid systems.

Also, automation and dependability are important features of third generation grids. The multitude and heterogeneity of resources in a grid will prevent humans to efficiently manage resource aggregation and usage. Third generation grids must be able to discover, bind, and use a great number of geographically distributed resources. Thus, automated configuration capabilities based on widely available metadata must be realized. Also, the multitude and distribution of grid resources make failures to happen so frequently that it can be though as the rule, not the exception. To cope with this, applications must be able to recover automatically from malfunction. Also, changes in the status of grid resources are so prevalent that automated reconfiguration capabilities are needed. All of these capabilities can only be implemented whether pervasive and robust information services are supported, which can provide detailed information about grid resources and their current status. Some of these aspects are closely related to the IBM's envision of 'autonomic computing' [28]. This term alludes to the autonomic nervous system, which is capable to control heart beats and body temperature, freeing the conscious brain dealing directly with these and other important low level functions. Thus, third generation grid systems need self-organizing, self-healing, and self-optimization capabilities to free applications to concentrate in the business logic.

3. GRID COMPUTING IN E-BUSINESS, E-COMMERCE AND E-GOVERNMENT

In this section we describe some services that can be made available thorough the grid, and that will be fundamental for the deployment of the next generation of e-business, e-commerce and e-government applications. These services provide integration of different types of resources in a level of transparency beyond the capabilities of current technology. Examples of such services are: Database access and integration, Knowledge discovery services (Data mining), distributed transaction services, and several other services which can take advantage of features of third generation grid computing platforms, such as automated reconfiguration capabilities. In this section we will focus mainly in two services: database access and integration and knowledge discovery.

In the following we discuss the current status of the research related to the implementation of these services using the grid, and the advantages of the grid as the platform of choice for the deployment of such services.

3.1 Database Access and Integration on the Grid

One service of major importance for many e-business, e-commerce and e-government applications is a virtual, uniform access to a set of heterogeneous, geographically dispersed databases. The fundamental value proposition for a Grid in general, and a database access service in particular, is virtualization [14], or transparent access to dispersed data sources, For data intensive applications, the following set of transparencies are relevant [10]:

Heterogeneity Transparency - An application accessing a data source should be independent of the actual implementation of that source, so that both can evolve independent of each other.

Name Transparency - with grid sources, applications should not even *specify* data objects explicitly. Instead, data access should be via *logical domains,* qualified by *constraints on attributes* of the desired object.

Ownership & Costing Transparency - As far as possible, applications should be spared from separately negotiating for access to individual sources located on different domains, whether in terms of access authorization, or in terms of resource usage costs.

Parallelism Transparency- An application processing data on a grid should automatically get the benefits of parallel execution over grid nodes.

Distribution Transparency - Applications should be able to maintain distributed data in a unified fashion.

The level of transparency described above is far beyond the capabilities of the current data management technologies. Recently several sets of services for integrating databases into the grid have been proposed recently [39, 10,9]. The objective is to provide at least a subset of the level of transparency just described.

The OGSA platform is a natural candidate for the deployment of such services. For instance, the nature of the Grid offers interesting new possibilities for areas such as distributed query processing [39]. Once a query has been compiled, distributed grid resources can be acquired on demand for running the distributed query execution middleware. The choice of which resource to allocate can be made based on parameters such as performance and price. Once the acquired resources are no longer necessary, soft state control can be used to automatically deallocate resources, making them available for other queries. In fact, there is currently a great deal of effort to propose and implement a distributed database access and integration service based on the OGSA and OGSI specifications.

In the international area, efforts towards this objective are concentrated in the Global Grid Forum [24] and on the OGSA-DAI project [32], the former financed by the British government.

3.2 Data Mining on the Grid

Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [23]. Data Mining (DM) is a step in this process that centers on the automated discovery of new facts and relationships in data. DM consists of three basic steps: data preparation, information discovery and analysis of the mining algorithm output [4]. All these steps exploit huge amounts of data and are computationally expensive. In this sense, several techniques have been proposed to improve the performance of DM applications, such as parallel processing [27] and implementations based on cluster of workstations [3].

Some recent work suggest that grids are natural platforms for developing high performance data mining services [7,27,33]. More specifically, Orlando et al. [33] describe the application of two data mining algorithms in the Knowledge Grid architecture [7]: the DCP Algorithm and the K-means Algorithm. However, other Data Mining techniques can take advantage of a grid infrastructure, as described, for instance, in [27].

3.3 Example of Utilization: an e-government Scenario

In this subsection we present a possible utilization scenario of the services just described, adapted from [10]: A government agency periodically mines patient records from various hospitals in a geographic defined area to detect biohazards. The patient records will be located in several, geographically dispersed, and heterogeneous databases. A grid-enabled database access service can deliver a union of the related records from those hospitals, qualified by symptom, for instance. This set of records will be used as input to a knowledge-discovery service on the grid that will mine the records, looking for patterns that would indicate the occurrence of a biohazard, such as the contamination of a river.

3.4 Autonomic Computing and the Grid

As mentioned in section 2.3, autonomic computing capabilities comprise important requirements which are driving the evolution of third generation grids. In this evolution scenario, grid services may be implemented as autonomic components capable to interact to each other [35]. Each service may be specified by a set of output services it can provide, a set of input services and resources it requires, and a set of policies to control its behavior and interaction to other services. In this scenario, self-configuration capabilities can be supported to simplify and reduce human interaction in configuration, version control, and maintenance tasks for complex e-business and e-commerce complex systems. Self-optimization capabilities allow services to be provided with better quality, by optimizing usage of resources and other services required. Also, service unavailability may represent significant loss for companies. By means of self-healing capabilities, a grid service could find a substitute for a required service or resource which has become unavailable, and reconfigure itself to go on working. In addition, self-protection capabilities can allow large and complex systems to self-defend automatically against malicious attacks and cascade failures. By taking self-healing measures and using early warning, systems can prevent from system wide failures.

Autonomic computing did not start with grid computing. However, many commonalities have been identified by their related communities. And finally, these new capabilities and functionalities which can be achieved by the cooperation between these two communities may significantly benefit e-commerce, e-business and e-government applications.

4. CONCLUSIONS

In this paper we presented fundamental concepts related with grid computing and the evolution of the grid as a platform aimed at the integration of resources at different levels. We discussed some services that can be made available through a grid platform and the importance of such services in an infra-structure capable of delivering the quality of service needed for e-business, e-commerce and e-government applications in the 21st century. We believe that while technology evolves it will become clear that other kinds of services will have the grid as the platform of choice for deployment, with the development of capabilities related to third generation grid computing platforms.

BIBLIOGRAPHY

1. Avaki, http://www.avaki.com
2. Baker M., Buyya R., Laforenza D. Grids and Grid Technologies for wide area distributed computing. Software – Practice and Experience 2002;32:1437-1466.
3. Baraglia R., Laforenza D., Orlando S., Palmerini P., Perego R., Implementation Issues in the Design of I/O Intensive Data Mining Applications on Clusters of Workstations. Proceedings of the 3rd Workshop on High Performance Data Mining, IEEE IPDPS'2000; Cancun, Mexico. Los Alamitos: Computer Society Press, 2000.
4. Berman F. From Teragrid to knowledge grid. Communications of the ACM 2001;44(11):27-8.
5. Bigus, Joseph P. *Data Mining with Neural Networks.* USA, McGraw-Hill, 1996.
6. Berendsen H. J. C., van der Spoel D., van Drunen R. GROMACS: A message-passing parallel molecular dynamics implementation. Comp. Phys. Comm.1995;91: 43–56.
7. Canataro M., Talia D. The Knowledge Grid. Communications of the ACM 2003;46(1):89-93.
8. Clark D. Face-to-Face with Peer-to-Peer Networking. Computer 2001;34(1):18-21.
9. Database Access and Integration Services Working Group: http://www.cs.man.ac.uk/grid-db/
10. DAIS Working Group. Services for Data Access and Data Processing on Grids. Available at http://www.gridforum.org/Meetings/ggf7/drafts/DAIS_DataServices.pdf
11. De Roure D., Jennings N.R., Shadbolt N.R. "The Semantic Grid: A Future e-Science Infrastructure". In *Grid Computing: Making the Global Infrastructure a Reality,* Berman F., Fox G., Hey T. Eds. Wiley Series in Comm. Networking and Distrib. Syst. West Sussex: John Wiley& Sons, 2003.
12. De Roure D., Baker M.A., Jennings N. R., Shadbolt N.R. "The evolution of the Grid". In *Grid Computing: Making the Global Infrastructure a Reality,* Berman F., Fox G., Hey T. Eds. Wiley Series in Communications Networking and Distributed Systems. West Sussex, England: John Wiley& Sons, 2003.
13. Douglas Thain, Todd Tannenbaum, and Miron Livny, "Condor and the Grid", in Fran Berman, Anthony J.G. Hey, Geoffrey Fox, editors, Grid Computing: Making The Global Infrastructure a Reality, John Wiley, 2003.
14. Ellis C., Gibbs S., Rein, G. Groupware. Some issues and experiences. Communications of the ACM 1991;34(1):38-58.
15. The FAFNER Web Page http://www.npac.svr.edu/factoring.html
16. Fayyad U.M., Shapiro G.P., Smyth P. "From Data Mining to Knowledge Discovery: An Overview". In: Advances in Knowledge Discovery and Data Mining, Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R., Eds.. MIT Press, 1996.
17. I.Foster. What is the Grid? A Three Point Checklist. *GRIDToday,* July 20, 2002. Retrieved October 2003 from http://www.gridtodav.com/02/0722/100136.html
18. Foster I., Kesselman C., Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomp. Applications 2001; 15(3), 2001.
19. Foster I., Geisler J., Nickless W., Smith W., Tuecke S. Software infrastructure for the I-WAY high performance distributed computing experiment. Proceedings of 5th IEEE Symposium on High Performance Distributed Computing, 1997. Los Alamitos: IEEE Computer Society Press, 1997.
20. Foster I., Kesselman C. (editors). *The Grid: Blueprint for a New Computing Infrastructure.* Morgan Kaufmann Publishers, 1999.
21. Foster I., Kesselman C. Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputer Applications 1997;11(2):115-128.

22. Foster I., Kesselman C., Nick J., Tuecke S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Presented at Global Grid Forum, June 2002. (Online Version http://www.gridforum.org/ogsi-wg/drafts/ogsa_draft2.9_2002-06-22.pdf)
23. Freitas A.A., Lavington S.H., *Mining Very Large Databases with Parallel Processing.* Kluwer Academic Publishers, 1998.
24. Global Grid Forum Web Page: http://www.gridforum.org/
25. The Globus Alliance Web Page: http://www.globus.org/
26. Grimshaw, A. and Wulf, W.. Legion: The next logical step toward the world-wide virtual computer. Communications of the ACM 1997;40(1):39–45.
27. Hinke H., Novotny J. Data Mining on NASA's Information Power Grid. Proc. HPDC 2000; Pittsburgh, USA. Los Alamitos; IEEE Computer Society Press, 2000.
28. Kephart J.O., Chess D.M. The vision of autonomic computing. Computer 2003;36(1):41-50.
29. Litiu R., Prakash A. Developing adaptive groupware systems using a mobile component framework. Proceedings of the ACM 2000 Conf. on Comp. Supported Cooperative Work (CSCW'00); December 2-6 2000; Philadelphia USA. New York: ACM Press, 2000.
30. Litzkow M., Livny M., Mutka M. Condor–A Hunter of Idle Workstations. Proceedings of the **8th** International Conference of Distributed Computing Systems (ICDCS'88). Los Alamitos: IEEE Computer Society Press, 1988.
31. Magowan J. A view on relational data on the grid. Proceedings of IEEE IPDPS'03. 2003 Apr 22-26; Nice. Los Alamitos: IEEE Computer Society Press, 2003.
32. Open Grid Services Architecture Data Access and Integration. http://www.ogsa-dai.org.uk/
33. Orlando S., Palmerini P., Perego R., Silvestri F. Scheduling high performance data mining tasks on a data grid environment, Proceedings of Intl. Conf. Euro-Par 2002. 2002 August 27-30; Paderborn, Germany. LNCS 2400. Springer-Verlag, 2002.
34. Palen L., Salzman M., Youngs E. Going wireless: behavior & practice of new mobile phone users. Proceedings of the ACM 2000 Conf. on Comp. Supported Cooperative Work (CSCW'00); December 2-6 2000; Philadelphia USA. New York: ACM Press, 2000.
35. Pattnaik P., Ekanadham K., Jann J. "Autonomic computing and the Grid". In *Grid Computing: Making the Global Infrastructure a Reality,* Berman F., Fox G., Hey T. Eds. Wiley Series in Communications Networking and Distributed Systems. West Sussex, England: John Wiley& Sons, 2003.
36. Platform Computing Corp. at http://www.platform.com
37. Rajasekar A.K., Moore R.W. Data and metadata collections for scientific applications. Proceedings of the European High Performance Computing Conference 2001. Amsterdam, Holland, 2001.
38. Tuecke S., Czajkowski K. , Foster I., Frey J., Graham S., Kesselman C., Snelling D., Vanderbilt P. Open Grid Services Infrastructure (OGSI), 2003. (Online Version http://www.gridforum.org/Meetings/ggf7/drafts/draft-ggf-ogsi-gridservice-23_2003-02-17.pdf)
39. Watson, Paul "Databases and the Grid". In *Grid Computing: Making the Global Infrastructure a Reality,* Berman F., Fox G., Hey T. Eds. Wiley Series in Comm. Networking and Distrib. Syst. West Sussex, England: John Wiley& Sons, 2003.
40. World Wide Web Consortium: http://www.w3.org/
41. http://www.sun.com/software/gridware/
42. http.//www.openpbs.org

43. Web Services Flow Language (WSFL) Version 1.0. IBM. Available at: http://www-4.ibm.com/software/solutions/Webservices/pdf/WSFL.pdf
44. Web Services for Business Process Design, http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm
45. Web Services Modelling Framework, http://www.cs.vu.nl/~dieter/wese/
46. ZHOU S. LSF: load sharing in large-scale heterogeneous distributed systems. Proceedings of the IEEE Workshop on Cluster Computing 1992. Los Alamitos, IEEE Computer Society Press, 1992.