An Advanced Grid Computing Course for Application and Infrastructure Developers

Erik Elmroth[1,2], Peter Gardfjäll[1] , and Johan Tordsson[1]

[1]Dept. of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

[2]HPC2N
Umeå University
SE-901 87 Umeå, Sweden

{elmroth, peterg, tordsson}@cs.umu.se

Abstract

This contribution presents our experiences from developing an advanced course in Grid computing, aimed at application and infrastructure developers. The course was intended for computer science students with extensive programming experience and previous knowledge of distributed systems, parallel computing, computer networking, and security. The presentation includes brief presentations of all topics covered in the course, a list of the literature used, and descriptions of the mandatory computer assignments performed using Globus Toolkit 2 and 3. A summary of our experiences from the course and some suggestions for future directions concludes the presentation.

1. Introduction

So far, the emphasis for Grid computing education has been on courses for *using* Grid technology, or using specific Grid middleware. Often, these courses are intended for students and researchers in application areas where Grid technology is likely to become important in the near future. For some previous courses, see [16, 31, 34].

Our focus is on an advanced course for the computer science students that soon will take part in the *development* of the future Grid infrastructure. This contribution presents our experiences from giving such a course during the spring of 2004.

Our objective was to develop an advanced Grid computing course for experienced computer science students with deep knowledge in many Grid-related fields. By defining the prerequisites accordingly, we were able to offer a course that focused on Grid-specific issues and that to a large extent relied on the fact that the students were familiar with much of the underlying technology. We also took advantage of the fact that they were experienced programmers.

The course covered all major parts of the Grid computing area, including architecture, programming, resource management, information infrastructures, security, data management, applications, Open Grid Services Architecture (OGSA), Open Grid Services Infrastructure (OGSI), Web services, high-throughput computing, portals, visualization and Virtual Reality (VR) on the Grid, sample Grid projects, and various Grid middleware. In computer assignments the students explored all major components of Globus Toolkit 2 (GT2) and gain experience in Grid service development in Globus Toolkit 3 (GT3) [25].

As the course was held for the first time, feedback from the students was important. In order to improve this dialogue, we made three questionnaires during the course; one at the first lecture, one in the middle of the course, and one after the course was finished.

In the following, we briefly present the topics included in the course. The assignments are covered in more detail, as they are of particular interest for instructors planning similar courses. This material is complemented with a description of the environment in which the assignments were performed, some reflections on the results of the questionnaires, and some suggestions for future versions of the course.

2. Student background

The course was primarily intended for fourth-year computer science students, but it also turned out to attract a few PhD students. In total, 32 students attended the course. The explicit prerequisites included the two courses "Distributed Systems" and "Parallel Computer Systems", which in turn had additional prerequisites, such as the courses "Computer Networks" and "Systems Programming".

0-7803-9074-1/05/$20.00 ©2005 IEEE

3. Course topics

In this section, we present the different topics covered by the lectures. Notably, most topics correspond to two hours of lectures. The topics are presented in the order they were scheduled.

3.1 Grid introduction

The first topic gave an introduction to the Grid computing area and introduced fundamental concepts and definitions. The introductory material included a general Grid computing overview, a Grid characterization, and a motivation for why the Grid revolution is taking place right now. Moreover, Grid usage was illustrated by some application scenarios; presentations of sample application projects; and Swedish, Nordic, and European infrastructure initiatives with local impact.

Concepts and definitions covered included a Grid resource definition, definitions of and relations between the concepts of protocol, Application Programming Interface (API), and software development kit, as well as their implications for portability and interoperability. Key issues for establishing a Grid infrastructure were identified, such as resource discovery, resource allocation, data management, security, accounting, monitoring, scalability, and fault-tolerance.

Study material: [2, 3, 10].

3.2 Grid architecture

Grid architecture was studied in order to establish a common view and vocabulary, and to understand how different types of components are combined to form a Grid infrastructure. The topic included characterizations and illustrative examples of the protocols in a layered Grid architecture, i.e., the fabric (interfaces to various resources), connectivity (such as communication and security), resource (for Grid-access to individual resource types), collective (for Grid-wide services), and application layer.

Study material: [8].

3.3 Grid programming

The aim of this topic was to bring insight into Grid-specific programming problems. The content included discussions on application considerations, presentations of sample projects, and software that illustrate common Grid programming techniques.

Application programming issues and characteristics discussed included performance, scalability, reliability, heterogeneity, application flow, serial vs. (trivially) parallel jobs, batch vs. interactive jobs, environment dependencies, checkpoint and restart, distributed data management, usability, etc. Sample usage scenarios highlighted the diversity of Grid applications, each with its own characteristics and software requirements.

Projects and software presented included MPICH-G2 [26], GrADS [18], Condor [12], Legion [27], NetSolve [17], and the Network Weather Service.

Study material: [10, 14].

3.4 GT2 overview

GT2 was presented with the objective to provide a high-level overview, with a brief presentation of all major components without covering more details than required for outlining their respective roles and interactions in a typical configuration.

Starting from a general protocol-oriented view of Grid architecture, the GT2 components were presented, including the Grid Security Infrastructure (GSI), Grid Resource Allocation Manager (GRAM), Resource Specification Language (RSL), Grid Resource Information Service (GRIS), Grid Index Information Service (GIIS), GridFTP, Globus Access to Secondary Storage (GASS), and the Replica Catalog.

Moreover, the GT2 portability and convenience API globus_common was introduced, as well as the connectivity API globus_io. Finally, some basic Globus design decisions and conventions were presented, including language selection, and naming schemes.

Study material: [2, 10].

3.5 Resource management

The resource management topic focused on the general problem of enabling secure, controlled, remote access to heterogeneous computational resources and the management of remote computations. Significant parts of the material covered the resource management components in GT2, and additional material extended the architecture with higher-level services for resource brokering and advance reservations.

The GT2 resource management architecture was presented in detail, illustrating all required interactions between a client, an information service, and the GRAM gatekeeper and job manager, as well as the interaction between GRAM and the local resource manager. The state model for GRAM jobs and the GRAM API were also presented, with illustrations of a range of common interactions for job submissions, status requests, state change callbacks, etc. The DUROC support for co-allocation was briefly presen-

ted. The RSL was described in some detail together with the globus_rsl module for RSL manipulation.

The resource brokering problem was presented in general terms and illustrated by the resource broker included in the NorduGrid ARC software [28]. Features such as advance reservations and co-allocation were discussed and illustrated by the GARA project.

Study material: [2, 4, 10].

3.6 Information infrastructure

Another major course topic was infrastructure solutions addressing the overall Grid information problem. The problem was introduced by identifying various types of static and dynamic information that are required by different Grid components, such as resource brokers. The Globus MDS-2 infrastructure was introduced and related to its less scalable predecessor, MDS-1. The behavior specified by the LDAP-based Grid Resource Information Protocol (GRIP) and Grid Resource Registration Protocol (GRRP) were briefly presented, including specifications for discovery, periodic notification, automatic directory construction, etc. The hierarchical MDS-components GRIS and GIIS were covered and exemplified by the local installation used for this course and the multi-level infrastructure used in the NorduGrid testbed. Moreover, the GRIS object hierarchy and its corresponding schema were presented.

Finally, the use of LDAP was exemplified by searching and filtering commands for finding the Grid resource information typically required.

Study material: [1, 2, 10].

3.7 Grid security

A wide range of Grid security concepts was discussed, as well as the Globus GSI. It should be emphasized that the students already were familiar with security fundamentals. Topics such as Secure Shell/Secure Socket Layer (SSH/SSL), encryption, Public Key Infrastructure (PKI), X.509 certificates, and mutual authentication, had been covered on previous courses. Hence, most of the time was spent on Grid-specific security requirements.

Basic security terminology was reviewed, which in addition to the concepts mentioned above included, e.g., authentication, authorization, integrity, non-repudiation, credentials, and trust domains. Grid-specific security requirements in typical usage scenarios were identified, including single sign-on; the need for delegation of rights; limited and restricted proxy credentials; Grid to local user identity mapping; and on-demand creation of temporary user accounts. The security support provided by the Globus Generic Security Service (GSS) API was also covered.

In addition to these basic Grid security mechanisms, the Community Authorization Service (CAS) was extensively described, and compared to the Virtual Organization Management System (VOMS).

Study material: [2, 6, 7, 10].

3.8 Data management

The requirements for Grid data management were illustrated by high-energy physics applications and different solutions for data transfer and data replication were discussed.

The GT2 GASS support for performing remote I/O and file staging was presented, including its architecture, API, extensions to RSL, and remote cache management support. GASS use was exemplified by various commands for manipulating the GASS cache, performing file transfers, and configuring GASS servers. The GridFTP protocol and its GT2 implementation were also presented.

A data replication infrastructure was motivated and the Replica Catalog introduced. A data replication scenario was outlined, resembling the infrastructure of the EU DataGrid project, including a replica manager, a replica location server, a replica metadata catalog, and a replica optimization server that adapts its behavior to network and storage element performance.

Study material: [2, 9, 10].

3.9 OGSA, OGSI, and GT3

We presented OGSA and the paradigm-shift it represents, where focus has moved from a protocol-centric towards a service-centric view of Grid computing. The topic also covered the closely related OGSI specification and its GT3 reference implementation.

Since OGSA is based on Web services, a brief description of Web services was given, and we also identified the missing pieces of functionality - statefulness and transience - added in OGSA in order to adapt Web services to Grid use.

Following the OGSA presentation, the fundamental concept of a Grid service was explained, covering the nature and capabilities of Grid services, as well as the core set of Grid service porttypes defined in the OGSI specification.

A brief component overview of GT3 was also given, including an explanation of how GT2 protocols map onto GT3 services.

Finally, we briefly touched the refactoring of the OGSI specification into the Web Services Resource Framework (WSRF), covering the motivation for the transition as well as its consequences. We summarized the main differences between the frameworks and introduced the WSRF-based Globus Toolkit 4 (GT4).

Study material: [5].

3.10 Grid service development in GT3

This topic gave a more practical perspective on OGSA and OGSI, in particular the development and deployment of Grid services using the tools and service primitives provided by GT3. The topic also served as an introduction to the third assignment.

As Web services make extensive use of XML and XML Schema, a crash course covering these topics was included. Moreover, a brief tutorial was given on how to build GT3 services using the Java-based tool Ant.

The remainder of the topic illustrated how to develop and deploy Grid services. A simple service example of "hello world" type was used to illustrate the Grid service development process, which for a simple service was summarized in the following five steps: defining the service interface; writing the service implementation; writing a deployment descriptor; building the service, and deploying the service in the runtime environment.

Study material: [10].

3.11 High-throughput computing

The purpose of this topic was to give an overview of high-throughput computing and Condor, with an emphasis on Grid applications.

The concept of high-throughput computing was presented and related to high-performance computing, a concept all students were acquainted with. Condor and basic Condor concepts, including universes and ClassAds were introduced. A more technical discussion of a typical Condor installation, including its daemons, pool layout, and the job startup procedure was covered. An overview of workflow management with DagMan, and Condor's data access mechanism was also included.

The final and major part was devoted to Grid-enabling Condor mechanisms. The benefits and limitations of Flocking, the Globus universe and GlideIns were discussed.

Study material: [12].

3.12 Visualization and VR on the Grid

The topic introduced Grid applications in the areas of visualization, VR and computational steering. Aspects of the required data transfer in relation to network bandwidth and latency were discussed. Trade-offs between sending raw data and complete rendered pictures were investigated, and parallel rendering algorithms were analyzed. Finally, some Grid-based projects were presented, such as ASCI, EU CrossGrid, Access Grid, VRVS, GameGrid, and RealityGrid.

Study material: [13].

3.13 Grid portals

The aim of this topic was to give some insight into the area of Grid portal development. The topic included both presentations of existing portal projects, and an introduction to some techniques commonly used when building a Grid portal. Portals were classified as general or application portals. NPACI Hot Page [35], the HPC2N Grid Portal [22], and the Grid Resource Broker [15] exemplified the former category. Application portals were illustrated by the ECCE environment [33], the Biology workbench [19], and the SLICOT Web computing environment [21].

Moreover, the MyProxy solution for managing a credential repository was presented, as it is frequently used in portal development.

Study material: [10, 11].

3.14 Local Grid research projects

Finally, the local Grid research projects at Umeå University were presented. In addition to the HPC2N Grid Portal and the SLICOT Web computing interface already mentioned, these were the SweGrid Accounting System (SGAS) [20, 32]; a resource broker [23], initially targeted for NorduGrid ARC; and semi-automatic tools for generating Grid computing interfaces to numerical software libraries.

4. Assignments

The course included three mandatory assignments: an introductory tutorial on the use of basic Globus tools, a large programming project where students implemented a system for solving master-worker applications using the Grid, and a smaller programming assignment giving the students hands-on experience of Grid service programming using GT3. Students were encouraged to work in pairs on each assignment.

4.1 Local Grid environment

The assignments were solved in a Grid environment distributed over two administrative domains, the Department of Computing Science and HPC2N. The latter provided two Linux clusters, one with 100 CPUs and the other with 240. The Department of Computing Science configured two resources, each with a batch system with eight computers as backend. Students also used these machines to compile and debug their programs. Each of the four resources ran a GRIS which registered with a local GIIS server.

The domains used different local security mechanisms, UNIX passwords and Kerberos tickets, which complicated local file system access. On the other hand, all resources had the same architecture (x86) and ran the same Linux operating system, thus avoiding possible compatibility problems.

The software used for the first two assignments were GT2.4, configured with the OpenPBS [29] batch system. Furthermore, a PostgreSQL [30] database was used for data persistency in assignment two, and GT3.0.2 Core was installed in each student's home directory for assignment three. In addition to the course literature, the Web information in Section 7 was recommended for solving the assignments.

4.2 GT2 tutorial

The first assignment followed a tutorial format, where the students used a computer to solve several small exercises. An instructor was available to answer questions. The purpose was to make the students familiar with basic GT2 tools. It was furthermore anticipated that knowledge of these tools would simplify the upcoming Grid programming assignments.

The basic GT2 tools for certificate and proxy management, information retrieval, job submission, and data transfer were used. Each tool was introduced by a series of examples. The students were encouraged to try out the tools, and for each tool solve exercises that required more advanced usage.

4.3 A fault-tolerant high-throughput PSE

The second assignment was a large programming project, where the students developed a Problem Solving Environment (PSE) for embarrassingly parallel applications using the Grid. The overall purpose was to gain practical experience of Grid programming and to discover the possibilities and difficulties associated with Grid infrastructure development. The assignment covered a broad range of GT2 protocols, including information retrieval, job submission, data transfer, and security. An additional purpose was to gain deeper understanding of topics such as resource brokering, scheduling, secure communication, and fault-tolerance.

The system was to solve the problem using a master-worker approach. The specific problem could be individually selected. Popular problems were Mandelbrot fractal generation and image ray-tracing.

Conceptually, the system was decomposed into three modules; a problem solving module (master), a node program (worker), and a resource broker.

The problem solver was to partition problems into subproblems, request appropriate resources from the resource broker, and submit jobs solving subproblems to the selected Grid resources. Once all subproblems were solved, the problem solver was to generate the solution to the overall problem from the solutions of the subproblems. The node program had to solve one instance of the problem on a Grid resource selected by the broker. Students were encouraged to use or modify an existing application for the node program.

The responsibilities of the resource broker module included to find available Grid resources, to classify which resources the node program could use, and to select resources to solve the subproblems. The broker was to base its decisions on the current load of the clusters and knowledge of the performance of previously submitted jobs. The broker was also responsible for staging files to and from the resources. See Figure 1 for an overview of the system components.

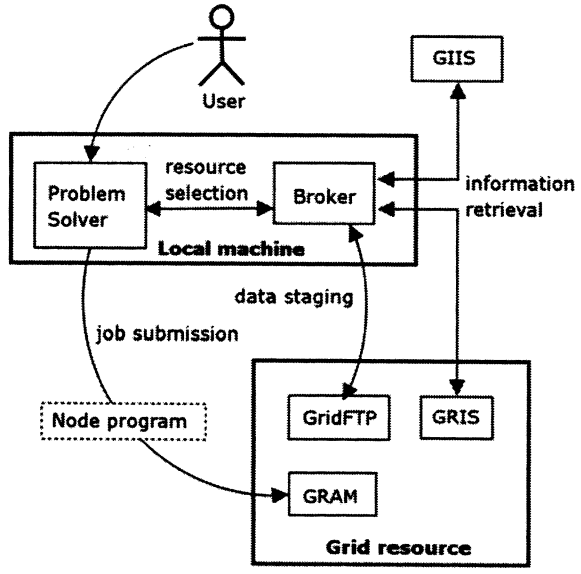

Figure 1. PSE component interactions.

As distributed systems in general are error-prone, fault-tolerance was emphasized. The implemented system was required to recover from failed node programs by resubmitting failed jobs, and from failed file transfers by restarting the failed transfers. If a file transfer failed repeatedly, the subproblem was to be submitted to another resource. File transfers failing partway through were to be resumed from the point of failure.

Moreover, the problem solver was required to recover from system shutdowns or crashes, and to handle multiple problem instances. If several uncompleted problem instances existed upon system startup, the user was to specify which one to continue to solve.

For examination, the students handed in a report describing the implemented system, including a speedup analysis. Students were also required to analyze the benefits and drawbacks of their resource selection algorithm.

In addition to the report, each pair of students did a 20-minute demonstration of their implementation and answered questions about their work.

4.4 An OGSA-compliant progress monitor

The aim of the third assignment was to provide the students with deeper insight into OGSA and OGSI, as well as to give hands-on experience of writing and deploying Grid services using GT3. The assignment also intended to highlight the paradigm-shift from the protocol-centric GT2 to the service-oriented GT3.

Since the second assignment intentionally was rather extensive we chose to limit the scope of the third assignment, in which a rather simple Grid service was to be implemented. The assignment extended the problem solver from the second assignment by publishing its state, giving users or applications push/pull-access to information about the progress of the problem solver, e.g., in terms of the amount of completed work.

The students were required to use at least one OGSI-compliant Grid service, for monitoring and publishing the problem solver progress, and a couple of service clients, one for push and one for pull-style access to the monitor service. The component interactions are illustrated in Figure 2.

Whenever a state change occurred in the problem solver, such as the completion of a job, a solver plugin was executed in order to notify the monitor Grid service of the state change. The monitor service notified its subscribers of the state change and also exposed its state through service data, for access by pull-style clients.

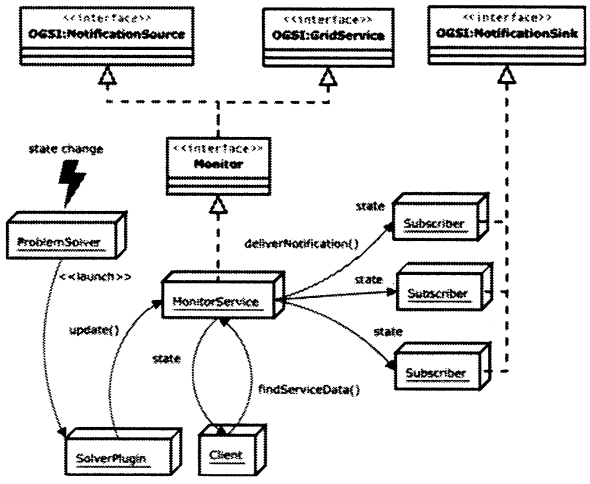

Figure 2. Progress monitor components.

5. Lessons learned

Based on our experiences from the course, the exam and assignment results, and the student feedback, our overall impression of the course is very positive. In total, 28 out of the 32 students passed the course. According to the answers of the questionnaires, the students' interest in the course subject had, in all cases, increased or remained unchanged during the course.

Our choice to put more focus on protocol-centric material rather than service-centric (i.e., GT2-related rather than GT3) was basically to ensure extensive coverage of material used in the assignments, and to perform the assignments in a well-tested environment. Judging from the student feedback, it appears that this probably should have been more explicitly motivated.

For future versions of the course, we plan to move the focus towards more service-oriented material. However, we will not make a full transition to WSRF and GT4 for the next offering of the course, but rather keep GT2 and GT3 for the assignments. The reason is that no stable and well-documented release of GT4 is available at the time of writing, and even if there was we would not have been able to acquire sufficient experience using it.

The feedback from the first assignment was very positive as it seems to perfectly have served the purpose of giving a quick hands-on introduction to the GT2 components used during the rest of the course. As this assignment was held early, directly after the GT2 overview lecture, some students felt that a comprehensive introduction to LDAP would have been beneficial.

For spring 2005, we plan to modify the last two assignments. In a migration towards more service-orientation, assignment three will be expanded at the

expense of assignment two.

Although the students have a strong background in programming, the second assignment proved to be very demanding. This was partly because the assignment appeared to be too extensive. Some Grid infrastructure problems also contributed to the difficulties. When reducing the total workload on this assignment, we will keep as much of the Globus programming as possible, and rather reduce the requirements for problem solver persistency and fault-tolerance.

Most students were pleased with the third, service-oriented, assignment. Some students also expressed that they would prefer a larger assignment on service-based architectures and a stronger overall course focus on Web services and GT3.

The Grid service development model, including writing WSDL files and deployment descriptors, generating stubs, etc, can be quite confusing for GT3 beginners. This was also expressed by some of the students, who had a hard time grasping the GT3 development process. We believe that this issue should be addressed by spending more time on presenting not only the GT3 components and their relationships, but also the service development model.

6. Concluding remarks

This contribution presents the material and our experiences from an advanced Grid computing course for experienced computer science students, held in the spring of 2004. One remark, in addition to the lessons learned (presented in Section 5) is that it was a great pleasure to give this course, and that much of its success was due to the great enthusiasm and hard-working spirit shown by the students. Undoubtedly, the students' specific interest in the area and previous experience have significantly contributed to making this an advanced course with a strong technical focus.

The course material for this and future versions of the course is available on the Web at [24] (mostly in Swedish). Notably, the specifications for the 2005 assignments are modified according to Section 5 and written in English.

7. Acknowledgements

We acknowledge the four anonymous referees for constructive comments.

8. Web sources used for assignments

- *GT2 GRAM*: http://www-unix.globus.org/api/c-globus-2.2/globus_gram_documentation/html/
- GT2 API: http://www.globus.org/developer/api-reference.html
- *RSL*: http://www-unix.globus.org/api/c-globus-2.2/globus_gram_job_manager/html/globus_job_manager_rsl.html
- *Understanding LDAP Design and Implementation - IBM redbooks*: http://www.redbooks.ibm.com/redbooks/ pdfs/sg244986.pdf
- PostgreSQL: http://www.postgresql.org/docs/
- Globus Toolkit Documentation Map: http://www-unix.globus.org/toolkit/docs/index.html
- The Globus Toolkit 3 Programmer's Tutorial: http://www.casa-sotomayor.net/gt3-tutorial/
- Open Grid Services Development Framework User's Guide:http://www-unix.globus.org/toolkit/3.0/ogsa/docs/java_programmers_guide.html
- Globus Toolkit 3 Core - A Grid Service Container Framework: http://www-unix.globus.org/toolkit/3.0/ogsa/docs/ gt3_core.pdf
- Open Grid Services Infrastructure (OGSI): http://www-unix.globus.org/toolkit/draft-ggf-ogsi-gridservice-33_2003-06-27.pdf
- GT3 Security Support: http://www-unix.globus.org/toolkit /3.0/ogsa/docs/security.html
- GT3 API: http://www-unix.globus.org/toolkit/3.0/ogsa/impl/java/build/javadocs/
- Apache Ant: http://ant.apache.org/

9. Course literature

[1] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid Information Services for Distributed Resource Sharing. In *Proc. HPDC-10*, IEEE Press, 2001.

[2] L. Ferreira, V. Berstis, J. Armstrong, M. Kendzierski, A. Neukoetter, M. Takagi, R. Bing-Wo, A. Amir, R. Murakawa, O. Hernandez, J. Magowan, and N. Bieberstein. *Introduction to Grid Computing with Globus*, Second edition, IBM Redbook Series, September 2003. (Chapter 1 - 7.)

[3] I. Foster. What is the Grid? A Three Point Checklist. *GRIDToday*, July 20, 2002.

[4] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, and A. Roy. A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation. *Intl Workshop on Quality of Service*, 1999.

[5] I. Foster, C. Kesselman, J. Nick, S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. *Global Grid Forum*, 2002.

[6] I. Foster, C. Kesselman, L. Pearlman, S. Tuecke, and V. Welch. The Community Authorization Service: Status and future. *CHEP03*, March 2003.

[7] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A Security Architecture for Computational Grids. *Proc. 5th ACM Conference on Computer and Communications Security Conference*, pp. 83-92, 1998.

[8] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 15(3), 2001.

[9] Globus. Getting Started with the Globus Replica Catalog. http://www.globus.org/datagrid/deliverables/replicaGettingStarted.pdf.

[10] B. Jacob, L. Ferreira, N. Bieberstain, C. Gilzean, J.-Y. Girard, R. Strachowski, and S. Yu. *Enabling Applications for Grid Computing with Globus*, First edition, IBM Redbook Series, June 2003.

[11] J. Novotny, S. Tuecke, and V. Welch. An Online Credential Repository for the Grid: MyProxy. In *Proc. HPDC-10,* IEEE Press, 2001.

[12] D. Thain, T. Tannenbaum, and M. Livny. Condor and the Grid. In F. Berman, et.al. (Eds.) *Grid Computing – Making the Global Infrastructure a Reality*, 2003.

[13] M. Thiebaux, H. Tangmunarunkit, K. Czajkowski, and C. Kesselman. Scalable Grid-based Visualization Framework, *Submitted to HPDC-13*, 2004.

[14] R. Wolski, N. Spring, and J. Hayes. The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing. *Journal of Future Generation Computing Systems*, 15, pp. 757-768, 1999.

10. Additional references

[15] G. Aloisio and M. Cafaro. Web-based access to the Grid using the Grid Resource Broker portal. *Concurrency Computat.: Pract. Exper.,* 14, pp. 1145-1160, 2002.

[16] A. Apon, B. Wilkinson, B. Ramamurthy, and J. Mache. IEEE Technical Committee on Scalable Computing: Education page. http://sol.cs.wcu.edu/~abw/TCSC/#Grid.

[17] D.C. Arnold, H. Casanova, and J. Dongarra. Innovations of the NetSolve Grid computing system. *Concurrency Computat.: Pract. Exper.,* 14, pp. 1457-1479, 2002

[18] F. Berman, A. Chien, K. Cooper, J. Dongarra, I. Foster, D. Gannon, L. Johnsson, K. Kennedy, C. Kesselman, J. Mellor-Crummey, D. Reed, L. Torczon, and R. Wolski. The GrADS Project: Software Support for High-Level Grid Application Development, *Int. J. High Performance Computing Applications*, 15:4, pp. 327-344, 2001.

[19] Biology Workbench. http://workbench.sdsc.edu/.

[20] E. Elmroth, P. Gardfjäll, O. Mulmo, and T. Sandholm. An OGSA-based Bank Service for Grid Accounting Systems. In *PARA'04. Workshop on State-of-the-art in Scientific Computing,* Lyngby, Denmark, June 2004.

[21] E. Elmroth, P. Johansson, B. Kågström, and D. Kreßner. A Web Computing Environment for the SLICOT Library. In *Proc. The Third NICONET Workshop*, pp. 53-61, 2001.

[22] E. Elmroth. M. Nylén, and R. Oscarsson. A User-Centric Cluster and Grid Computing Portal. *Submitted*, 2004.

[23] E. Elmroth and J. Tordsson. A Grid Resource Broker Supporting Advance Reservations and Benchmark-based Resource Selection. In *PARA'04. Workshop on State-of-the-art in Scientific Computing,* Lyngby, Denmark, June 2004.

[24] Grid: arkitekturer, programvaror och tillämpningar. http://www.cs.umu.se/kurser/TDBD20/.

[25] Globus. http://www.globus.org.

[26] N. Karonis, B. Toonen, and I. Foster. MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface. *Journal of Parallel and Distributed Computing (JPDC)*, Vol. 63, No. 5, pp. 551-563, May 2003.

[27] A. Natrajan, A. Nguyen-Tuong, M.A. Humphrey, M. Herrick, B.P. Clarke, and A.S. Grimshaw. The Legion Grid Portal. Concurrency Computat.: Pract. Exper., 14, pp. 1365-1394, 2002.

[28] Nordugrid. Advanced Resource Connector (ARC). http://www.nordugrid.org/middleware/.

[29] Portable Batch System. http://www.openpbs.org.

[30] PostgreSQL. http://www.postgresql.org.

[31] B. Ramamurthy. GridForce: A Comprehensive Model for Improving the Technical Preparedness of our Workforce for the Grid. *In Proc. CCGrid'04*, IEEE, to appear.

[32] T. Sandholm, P. Gardfjäll, E. Elmroth, L. Johnsson, and O. Mulmo. An OGSA-Based Accounting System for Allocation Enforcement across HPC Centers. *The 2nd International Conference on Service Oriented Computing (ICSOC04),* ACM, 2004.

[33] K.L. Schuchardt, B.T. Didier, and G.D. Black. Ecce - A Problem-Solving Environment's Evolution Toward Grid Services and a Web Architecture, *Concurrency Computat.: Pract. Exper.,* 14, pp 1221-1239, 2002.

[34] H. Stockinger, F. Donno, R. Puccinelli, and K. Stockinger. Data Grid Tutorials with Hands-on Experience. *In Proc. CCGrid'04*, IEEE, to appear.

[35] M. Thomas and J.R. Boisseau. Building Grid Computing Portals: The NPACI Grid Portal Toolkit. Texas Advanced Computing Center, Univ. of Texas at Austin.