

Grids: The top ten questions

Jennifer M. Schopf^a and Bill Nitzberg^b

^a*Mathematics and Computer Science Division,
Argonne National Lab, Department of Computer
Science, Northwestern University, USA*

E-mail: jms@mcs.anl.gov

^b*Veridian, PBS Products, USA*

E-mail: bill@computer.org

The design and implementation of a national computing system and data grid has become a reachable goal from both the computer science and computational science point of view. A distributed infrastructure capable of sophisticated computational functions can bring many benefits to scientific work, but poses many challenges, both technical and socio-political. Technical challenges include having basic software tools, higher-level services, functioning and pervasive security, and standards, while socio-political issues include building a user community, adding incentives for sites to be part of a user-centric environment, and educating funding sources about the needs of this community. This paper details the areas relating to Grid research that we feel still need to be addressed to fully leverage the advantages of the Grid.

Keywords: Grid computing, survey

1. Introduction

Grids are not a new idea. The concept of using multiple distributed resources to cooperatively work on a single application has been around for several decades. As early as the late seventies work was being done toward “networked operating systems” [13,23,65]. In the late eighties and early nineties, this returned as distributed operating systems [9,14,64]. Shortly thereafter the field of heterogeneous computing [22,41] came to play: thousands of heterogeneous resources running hundreds of tasks. The next derivative of this area was known as parallel distributed computing in which parallel codes ran on distributed resources. This became metacomputing, and then computing on the Grid [20].

Even with this legacy, there are several differences between today’s Grids and the older distributed operating system work.

- *Grids focus on site autonomy.* One of the underlying principles of the Grid is that a given site must have local control over its resources, which users can have an account, usage policies, etc.
- *Grids involve heterogeneity.* Instead of making every administrative domain adhere to software and hardware homogeneity, work on the Grid is attempting to define standard interfaces so that any resource speaking a defined set of protocols can be used.
- *Grids involve more resources than just computers and networks.* Grid computing today is as much about the data as it is about the compute cycles – where is the data, how to store terabyte data sets, how to replicate the needed pieces of data and access them through the network. Specialized scientific instruments, such as earthquake shake tables [47] and x-ray research facilities [2] are also being added into this picture.
- *Grids focus on the user.* This is perhaps the most important, and yet the most subtle, difference. Previous systems were developed for and by the resource owner in order to maximize utilization and throughput. In Grid computing, the specific machines that are used to execute an application are chosen from the user’s point of view, maximizing the performance of that application, regardless of the effect on the system as a whole.

It is these differences that create many of the problems presented below but which make the Grid a more usable system than its predecessors.

Yet today, the Grid is still miles away from being more than an academic concept. In a recent news story [15], it was wondered if the Grid was “merely an excuse by computer scientists to milk the political system for more research grants so they can write yet more lines of useless code”. There are many independent groups solving similar problems in ways that will not interoperate. And real users are few and far between, to the point that it has been argued that the Grid is a solution in search of a problem [59].

However, there is a large group of researchers currently working on various pieces of Grid technology, as detailed in the following sections. In addition, mem-

bers of the community have created the Global Grid Forum [25] which has the goal of being an arena where researchers from various groups and funding sources can interact with the hope of defining best-practice documents, API and protocol standards. It should also be noted that none of this work is standing still. Recent advances are addressed in several sections and in the conclusion. However, as much progress has been made, there is still much, much more work to be done.

This paper discusses where we feel that work is needed in order to make the Grid a reality, not only in terms of research but in socio-political terms as well. Technical challenges include having basic software tools, higher-level services, functioning and pervasive security, and administration, while socio-political issues include building a user community, adding incentives for sites to be part of a user-centric environment, and educating funding sources about the needs of this community. They are presented in no particular order.

2. Why don't Grids have basic functionality yet?

The vision of a Grid has been present for several years now. For computational Grids to be considered a success, according to one source [20], they should be pervasive, dependable, consistent and inexpensive. By combining these four properties Grids can have a transforming effect on how computing is performed and used, not only in terms of computation but data management as well.

However, before these goals can be addressed, basic functionality must be ensured. At this time, Grids are only rarely used as they were meant to be. By this we mean that application developers are not using multiple machines at geographically distant sites to coordinate and solve a single application [19]. Grids are being used to solve embarrassingly parallel applications (such as Seti@home [58], Entropia [17], Condor [10], etc.) or for easier resource selection (such as the PACI Genie work [24] or GRB [33]). The use of distributed resources as such is only seen in demos, supercomputing conferences, and special occasions, and still requires extensive coordination between sites. This is primarily because Grids do not yet have the full basic functionality needed for more extensive use.

One question to answer is what do we mean by basic functionality. A simple example to help define this is the scenario of what needs to be done by a user to start running an application over the Grid. Given an MPI code, for example, we would like to be able

to just run "make", and some command to run the job, "grid-run". To do this today, the first step is to determine what software may need to be installed on the machine(s), installing and then checking it. The security infrastructure needs to be in place as well, which means making sure all the security credentials are setup, not only certificates, keys, etc., but accounts as well. The next step generally involves trying to get information about the available resources – which means determining where to ask, what you can ask, and what the answer means. After determining where to run a job, it must then be submitted (which may include staging files, setting up the proper environment, etc.), run (preferably in a way that the progress can be monitored or adapted), and then cleaned up afterwards (including data movement and analysis). None of these are straightforward, single commands for a user to just run.

All of the pieces to do this exist today in one form or another. What doesn't yet exist is the entire service to do this in a simple and straightforward manner. The interoperability does not exist, nor do the higher-level tools that are needed. The few current users have had to go through heroic measures to achieve any functionality at all.

Getting a Grid to function correctly can be complicated by the fact that for any part of it to work, several other parts must function as well, which can be a socio-political problem. Cooperation between research groups in different areas can be difficult. In addition, because the various pieces of software or tools may need to be upgraded at the same time, there is resistance to this from both administrators (since getting one new piece of software up and running at a time is difficult enough) and users (who may have to learn an entirely new environment). Other socio-political issues involve the difficulty in getting funding to address basic functionality issues, as addressed in Section 8, as well as the need for variance management (see Section 5), deployment issues (see Section 6) and security (see Section 3).

Moral: Before we can have a successful Grid, we must have a functional Grid.

3. Why aren't there more Grid application developers and users?

Behind the development of Grid technology is a large set of application developers who are in dire need of the resources that a computational Grid will offer. Ar-

guments can be made as to how much the Grid can help both traditional computational scientists (biologists, chemists, physicists, etc.), and non-scientists as well (such as film distributors [5]). The computer scientists working in the area almost uniformly consult application scientists so that their work will aid the needs of this user base, which was also the motivation behind the development of an Applications User Working Group [27] as part of the Grid Forum. However, at the March 2000 meeting of the Grid Forum that group realized that while there is a need for user input to guide the research, those users are hard to find [4], and getting input is even more difficult (see the discussion of use cases in Section 2).

So the next question to ask is: Why is this the case? Both NSF-funded supercomputer centers (NPACI and NCSA) boast large user communities. Countless groups have used Grid resources to further their work. However, for the most part, these users have been running highly specialized codes that were targeted to specific platforms. Currently, most everyday scientists use the resources set up by various Grid developers (NCSA [45], NPACI [50], IPG [39], Condor [10], etc.) for access to a wider set of machines. These machines are then used individually and, for the most part, as specific machines. That isn't to say that these resources aren't a gain to the community, because they are. They simply aren't being used as a proper Grid.

In fact, making applications "Grid enabled" is seen by some as a distraction from getting real science done [43]. Many applications groups have tried to become more involved in Grid computing because of the lure of large funding programs (such as the ITR program from NSF and the SciDAC program through the Department of Energy). However, to meet these goals, it can be the case that other software development work is delayed. This further causes application scientists to wonder if the Grid isn't just a buzzword used to get funding, without the usability to back it up.

To have a true Grid user community, a number of issues need to be addressed. Better software tools are required to ease the transition to this new environment (see Section 2). Standards must be developed to supply a uniform interface to Grid services (see Section 4). Deployment of Grid software must be made easier (see Section 6). And basic functionality, as discussed in the previous section, must become the default, not the exception.

Moral: Without users there can be no Grid.

4. Where are the Grid software tools to aid application developers?

One of the lessons that was learned as part of the development of the parallel computing field is that the software aspects can be at least as hard as the hardware [66]. Until the development of debuggers, fast compilers, and other flexible tools, programming a parallel architecture was extremely difficult, and therefore few application scientists attempted it.

Several groups are actively working toward a basic set of software tools for the Grid. These include Globus [19,31], Legion [42], Condor [10], and Cactus [6] as well as several portals efforts [11,24,46], among many others. However, the learning curve for these tools is still steep. In addition, there is a socio-political problem in that for the most part, these are academic projects. There is little gain in academics to hardening code into a product, and maintaining a tool set is made more difficult by graduating students. Also, there is often little or no funding for hardening code (see Section 8). For these tools to be more widely accepted, this lack of priority must change.

Another aspect that highlights the difficulty in developing useful tools is the lack of well-defined use cases for Grid tools. This is a chicken and egg problem. In order to know what tools to develop, computer scientists need to know how the tools will be used. However, without tools to use, application scientists don't know what to ask for. This is also complicated by the difficulty in constructing a reasonable use case – it must have enough details to be specific enough, and yet not so many as to be either overwhelming or off topic. The difficulty in finding the right level to describe the use cases can be complicated by an application developer not knowing even what to ask for or not being able to define what is possible. From a socio-political point of view, a complicating factor can be determining simply who will construct the use cases, the computer scientist building the tool or the application developer using it. Also, the very different language and communication styles seen between computer science-centric projects and application-centric projects can hinder forward progress significantly. Communication between groups developing the Grid is a constant source of difficulty [53].

Moral: The Grid needs tools that facilitate the use of a Grid.

5. How do we make Grids secure?

The Grid will not be widely used unless it can be secure in terms of access, communication, and having encryption available. This is a technical problem that has gotten a lot of attention, but not as much research as needed. Every researcher realizes this, going back to the original NOW paper in 1994 [1,49]. And yet, for many systems, the mandatory “security is important” paragraph is all the depth the problem receives.

In the original version of this paper [61] we stated that although “work has been done to address the [need for security infrastructure] (including [work by] Legion [42] and Globus [3,21]), without agreement from all the sites on the Grid, it will not function. While much progress has been made in this area lately within the Global Grid Forum [29], this work is still not compatible with the larger community (aka Legion or Kerberos).”

In the 18 months since, a portion of this has been resolved. The Grid Security Infrastructure (GSI) [29, 63] has become an accepted security infrastructure, and is well on its way to becoming a standard through both the Global Grid Forum [30] and IETF [38]. It is operational over Kerberos as well as PKI, and many standard tools are being ported to use it, such as AFS [37], CVS [34], OpenSSH [35], and MyProxy [52], an online credential repository for enabling secure grid portals.

However, this only resolves the issue of authentication, or proving a user is who they say they are. Many higher-level issues, and indeed higher-level tools, are still left unresolved. For example, for a user to be able to use the standard GSI tools to access machines that are part of two different testbeds, arrangements must be made for their certificate authorities to recognize each other. Technically, this is simply resolved. Practically, this issue involves deep security decisions at sites, and is anything from straightforward [67].

Furthermore, the issue of authorization, or what the usage policy is, has not begun to be addressed in a uniform way. Even with a GSI-enabled infrastructure in place, a local account is needed to allow access to the resource. This is yet separate from the issue of accounting, discussed in Section 6.

Moral: Without a security infrastructure (including higher-level tools) users will not take advantage of the Grid.

6. How can we define standard interfaces and definitions for the Grid?

A new user to the Grid is likely to ask questions such as “How do I run a job on the Grid?”, “What sort of

monitoring is available?”, “Where do I get information about the Grid resources?”, or “How do I make sure this operation is secure?” Currently, there is no simple answer to any of these questions.

The lack of standards can be seen not only in the need for common protocols, where competing approaches are having a similar affect to that seen in pre-TCP/IP networking, or APIs, where for example MPI resolved message passing differences in that setting, but in simple language usage as well. One instance of this can be seen in the lack of an agreement among tool developers of basic definitions of terms. We all know that defining the term “job” or “resource” can lead to a several hour religious argument, but if a community standard were made, each group would be able to translate the meaning of the term in their dialect into a general lingua franca, thereby encouraging greater interoperability between current tools.

Another example can be seen with information services. There are two approaches gaining acceptance in this area. The first is the re-designed Globus Metacomputing Directory Service (MDS 2.1), which consists of two protocols [8] and the Globus reference implementation of them [44]. The second is the Grid Monitoring Architecture (GMA), developed by the GGF Performance working group [28,62], which has three reference implementations under development [7,12, 60], and being deployed as part of the European Data-Grid project [18]. These two approaches actually address different pieces of the monitoring problem; MDS concentrates on the resource discovery portion, while GMA concentrates on the provision of data. So one would hope they would interoperate, and work off of each others strengths, but the truth is far from it. There is currently little or no collaboration between these efforts, and in fact they may be viewed as developing quite different standards for interfaces, APIs, and protocols in a very overlapping space.

Contrary to this example, there have been some efforts to increase communication and standardization in the field. For example, standardization is one of the goals of the Global Grid Forum [25] in general. However, it is also a goal of the Peer-to-Peer Working Group [54], the New Productivity Initiative (NPI) [51], .net [48], and others. So a further question is who will standardize between the standardization bodies? The Economist points out “Once the commercial potential of the Grid begins to dawn, standard-setting skirmishes will break out.” [15]. It’s safe to argue they already have.

Moral: The lack of standards continues to hinder the interoperability needed for the Grid.

7. How can we manage variance on the Grid?

One of the primary difficulties in using the Grid is the unpredictable behavior of the resources. This is in part related to the lack of global control, discussed in the introduction, in that a user is never sure who else will be using a shared resource, or in what manner. The bandwidth over a given network link varies not only due to time-of-day usage, but due to individual application use, and quite widely in fact, up to three orders of magnitude in an hour [68]. Machine loads, queue lengths, and many other resources also have high variances in small time frames, leading to unpredictable behavior.

Unpredictable behavior affects applications in several ways. First, without information about the variance behavior of a resource, decisions regarding services, from fault management to resource selection, will have poor quality results. Second, there is the socio-political problem that results when a user has an application with varying performance. It has been our experience [55] that users want not only fast execution times from their applications, but predictable behavior, and would be willing to sacrifice some performance in order to have reliable run times.

Variance behavior needs to be taken into account in almost every aspect of Grid research. Monitoring tools can examine ways to predict and report perturbations in systems; information services must be developed to handle the additional information regarding variance; users must be taught how variance can affect their performance; scheduling algorithms should be developed to incorporate variance in the decisions being made. There is nothing we can do to decrease the performance variance of Grid resources, therefore we should find ways to take advantage of it.

Moral: Varying behavior is a fact in the Grid and must be addressed.

8. How can deployment be made easier?

One of the main unresolved issues for new application developers trying to expand to use Grid technology is the difficulty in deploying a testbed. Many groups get funding to start adaptation of their applications for use in a Grid environment, only to be stymied by the difficulty in basic installation and system administration issues. As an example of this, we have been told of numerous occasions when email has been sent to the leads of several large grid projects stating things like “I

tried to install your GridSoftwareX, but it proved rather difficult, so I decided to just write my own . . .” to, as one might imagine, less than satisfactory results.

The deployment of the NASA Information Power Grid (IPG) [39] began in the late 1990’s, and is only recently considered a production system by many involved. As part of this effort, Johnston developed guidelines for other Grid efforts to assist with testbed setup [40]. From this one can see the difficulties involved, not only technical, but socio-political. System administrators are used to having control over their resources, so any software that even appears to affect this, as many Grid software packages appear to, are seen as threatening. Firewalls, present in many settings, are also often a major stumbling block to deployment, again as much for socio-political issues as technical ones. Many of the current Grid projects are academic, and there is little academic gain in hardening software or packaging it well for users, unfortunately.

Another stumbling block in deployment is the difficulty in establishing an agreed upon accounting methodology [26]. In current systems, in order to use a resource a user will most likely need to set up an account on that resource. This involves dealing with many policies that were never constructed with the Grid in mind in terms of proving your identity and agreeing on acceptable usage policies. Due to both technical and socio-political problems, a halfway solution to this problem can be even worse than no solution at all [36]. However, getting account access to a resource is a fundamental need for any resource usage.

Moral: If the deployment and set up of software isn’t seamless, getting users to adapt their systems to be Grid compatible will not happen.

9. Where are the benefits to encourage sharing on the Grid?

One of the main differences between work on the computational Grid and previous distributed computing research is that Grid development has focused on the needs of the end user and the top-level policymakers, rather than focusing on the middle-level resource owners and managers. This has resulted in a clear motivation for Grids for both the end users and the policymakers, but not for the resource owners. From the end user’s perspective, Grids offer easy ways to leverage their access to resources that may be spread out both geographically and across many administrative domains, even when the managers of those resources

cannot agree either on technical or socio-political issues. At the highest levels, Grids have the promise of directly implementing enterprise-wide organizational policy. For example, NASA could direct 90% of all its resources to deflecting a near-Earth asteroid collision (regardless of location, etc given a functional Grid) – a compelling argument for all of us.

However, both these new capabilities lead to a new set of issues yet to be solved. First, this user-centric approach puts the responsibility on the user to “play nicely”. Sharing is hard. As seen in the tragedy of the commons, individuals act in their own best interests and tend to overuse and degrade any shared resource. For the most part, this problem existed before Grids. However, the introduction of the Grid concept has made it much easier to exploit the “commons”. For example, as tools develop, users will have the option of instead of selecting a single resource to run a job, for example, submitting it to a large number of sites, and then just discarding the results from all but the first to complete. This will give the user the fastest turn-around time, but will undoubtedly adversely affect many other users, as well as waste cycles for no reason. Therefore the question remains as to what benefit can the Grid offer to users that will keep them from being selfish?

Second, the middle-level resource owners and managers are caught, well, in the middle. Users are exploiting Grid technology to consume greater amounts of valuable computing resources, but most Grid accounting techniques are not yet capable of providing the controls and measures needed. In effect, users end up getting a free ride, and the middle-level resource owners have little to show how their individual sites are of benefit. Beyond the technical issues involved in deploying Grid technology, individual resource owners face the dual problems of controlling user’s consumption and proving their value to the policymakers. Without Grid-level accounting in place, what benefit can the Grid offer the people responsible for individual systems or sites?

Moral: Benefits to middle-level resource owners (and therefore buy-in from all levels of the management hierarchy) must be made clear in order for Grids to be deployed.

10. How can we fund the work needed for a functioning Grid?

As we’ve been stating, Grids need to have users to be successful. To have users, Grids must have basic

functionality, standards, well-developed tools and easy deployment. However, to do any work on the Grid, researchers need funding. Funding agencies are being put under a lot of pressure to fund only new and innovative projects. They have been funding these for Grid computing, as evidenced by the recent NSF ITR program, NASA’s programs on grid computing, and the DOE SciDAC program. However, much work on the Grid is viewed as mundane, for example hardening the software to the point of usability. The innovative part of working on the Grid is the coordination of sites and resources. However, it is often believed that this is not new enough, so work addressing that approach is often not funded, to the detriment of the would-be users, who then complain when they receive funding to “gridify” their codes when the basic tools aren’t there.

As another example, one of the best possible approaches that could be used in Grid research is to fund several projects to work on the same problem taking different approaches, and then to provide funding for them to be combined or for the strongest to leverage off of work by the others. In this way, the entire problem space of that problem would be likely to be explored. Furthermore, agreement between the research groups would show a trend to a best-uses policy. However, giving funding to multiple groups to do one thing can be viewed as a waste of resources so this is unlikely to occur. The tension between the need to find the best approach and the need to converge on a standard is also felt here. However, as seen with Microsoft, having one monolithic group defining the way things should be done is often a disadvantage to the very innovators that should be being encouraged.

Funding agencies need to be educated on the needs of Grid researchers. Money should be made available for the mundane tasks of getting software beyond beta-versions, for getting test-beds set up and functional, and for basic functionality work to be completed. Funding is also needed to support work toward community standards. Without this support, the Grid will not mature as it should.

Moral: Without support for basic work in functionality, standards and software engineering, the Grid will not live up to its potential.

11. Where are the performance metrics for success?

In this paper we have addressed a number of areas in that we feel more work is needed to have a truly

functional Grid. As with every large project, a success metric is needed to determine the quality of the result. We propose three:

1. The Grid can be considered a success when there are no more “Grid” papers, but only a footnote in the work that states, “This work was achieved using the Grid.”
2. The Grid can be considered a success when supercomputer centers don’t give a user the choice of using their machines or using the Grid, they just use the Grid.
3. The Grid can be considered a success when a SuperComputing [56] demo can be run any time of the year.

12. Conclusion

In the 18 months since the first draft of what had been many on-going conversations was put to paper, some progress in these areas has been made. In terms of basic functionality, daily usage of the Globus Toolkit has “proved it to be a robust standard” [15] for people other than Globus gurus. However, no one would disagree that what is there today is simply not enough. Similarly, there has been increased commercial interest and support for Grids [32], but one still cannot just buy a commercial solution to these problems, nor is that expected any time soon. The funding situations has improved, with large programs from NASA, NSF and DOE, and there has been more attention on the need for funding for testbeds and support, but again, this is far from resolved.

One interesting metric of progress is the shift in the “Top Ten” list presented in the original version of the paper [61], in a talk version given during the summer of 2001 [57], and this version. The need for better-defined use cases and simpler deployment has been strengthened, as has the need for basic information and basic information services. The advances in technology have also been seen throughout, but especially with respect to security, funding and basic functionality. The only point in the original work, which became a footnote in the talk-version and we no longer discuss here is the need for cost models. This isn’t because it has been resolved, but because it’s level of importance, when compared with the points contained here, is significantly less.

This paper identifies the key areas of research in order to foster the design and implementation of a na-

tional computing and data grid. These challenges are both technical and socio-political, but can be overcome once identified. We look forward to several years from now when the problems addressed here are considered archaic.

Acknowledgements

We gratefully acknowledge the many folks who have had rambling conversations at meetings, universities, mailing lists, and bars that fostered this paper. We would especially like to thank Alain Roy, Ian Foster, Steve Tuecke, Brian Tierney, the Grid Forum Scheduling Working Group, and all the UCSD AppLeS folks. This work was supported in part by the Mathematical Information and Computational Sciences Division Subprogram of the Office of Advanced Scientific Computing Research, US Department of Energy, under contract W-31-109-Eng-38, by NASA, and by NSF Awards ACI-00933000 and PACI#763-04/ACI-9619019.

References

- [1] T.E. Anderson, D.E. Culler and D.A. Patterson, A case for networks of workstations: NOW, *IEEE Micro* (February 1995).
- [2] Advanced Photon Source, National Synchrotron Radiation Research Facility, Argonne National Laboratory, Illinois, USA, <http://www.aps.anl.gov>, 2001.
- [3] R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer and V. Welch, A National-Scale Authentication Infrastructure, *IEEE Computer* **33**(12) (2000), 60–66, also available from <http://wwwwfp.globus.org/documentation/incoming/gsi-deploy.pdf>.
- [4] F. Berman, Personal Communication, 2000.
- [5] M. Beck and R. Moore, The internet2 distributed storage infrastructure project, *Computer Networking and ISDN Systems* (1998), 2141–2148.
- [6] Cactus, <http://www.cactuscode.org/>, 2001.
- [7] B. Coghlan, A. Djaoui, S. Fisher, J. Magowan and M. Oevers, Time, Information Services and the Grid, Proceedings of the British National Conference on Databases, 2001.
- [8] K. Czajkowski, S. Fitzgerald, I. Foster and C. Kesselman, Grid Information Services for Distributed Resource Sharing, *Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10)*, IEEE Press, August 2001.
- [9] D. Cheriton, The v distributed system, *Communications of the ACM (CACM)* **31**(3) (March 1988).
- [10] Condor: High throughput computing, <http://www.cs.wisc.edu/condor/>, 2001.
- [11] Computing portals, <http://www.computingportals.org/>, 2001.
- [12] B. Crowley, B. Tierney, D. Gunter, J. Lee and M. Stouffer, Python GMA, <http://sourceforge.net/projects/py-gma/>, 2001.
- [13] W.J. Donnelley, Components of a network operating system, *Computer Networks* **3**(6) (December 1979), 389–399.

- [14] F. Douglass and J. Ousterhout, Transparent process migration: Design alternatives and the sprite implementation, *Journal of Software-Practice & Experience* **21**(8) (August 1991).
- [15] The Economist, Computing Power on Tap, published in The Economist, June 21, 2001, also available from http://www.economist.com/science/tq/displayStory.cfm?Story_id=662301.
- [16] European DataGrid Work Package 3 – Grid Monitoring Services, <http://hepwww.rl.ac.uk/DataGridMonitoring/>, 2001.
- [17] I. Entropia, <http://www.entropia.com/>, 2001.
- [18] <http://hepwww.rl.ac.uk/DataGridMonitoring/>, 2001.
- [19] I. Foster and C. Kesselman, The Globus project: A status report, in *Proceedings of IPPS/SPDP '98 Heterogeneous Computing Workshop*, also available from <ftp://ftp.globus.org/pub/globus/papers/globus-hcw98.pdf>, 1998.
- [20] I. Foster and C. Kesselman, eds, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, Inc., 1999.
- [21] I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, A security architecture for computational grids, in *Proceedings of the 5th ACM Conference on Computer and Communications Security Conference*, also available from <ftp://ftp.globus.org/pub/globus/papers/security.pdf>, 1998.
- [22] R. Freund and H. Siegel, Heterogeneous processing, *IEEE Computer* (June 1993), 13–17.
- [23] H. Forsdick, R.E. Schantz and R.H. Thomas, Operating systems for computer networks, *Computer* **11**(1) (January 1978), 48–57.
- [24] Grid enable interactive environment-genie, <https://genie.paci.org/>, 2001.
- [25] Global Grid Forum, <http://www.gridforum.org>, 2001.
- [26] Grid forum account management working group, <http://www.nas.nasa.gov/bthigpen/accounts-wg>, 2001.
- [27] Global Grid forum application working group, <http://dast.nlanr.net/GridForum/Apps-WG/>, 2001.
- [28] Global Grid Forum Performance Area, <http://www.gridforum.org/4-GP/Perf.htm>, 2001.
- [29] Global Grid Forum Security Area, <http://www.gridforum.org/security/>, 2001.
- [30] GGF Security Area, Grid Security Infrastructure Working Group, <http://www.gridforum.org/security/gsi/index.html>, 2001.
- [31] Globus, www.globus.org, 2001.
- [32] 12 Companies Adopt Argonne Lab/USC Globus ToolkitTM as Standard Grid Technology Platform, <http://www.globus.org/developer/news/20011112a.html>, 2001.
- [33] <http://sara.unile.it/grb/grb.html>, 2001.
- [34] <http://www.globus.org/gridcv/>, 2001.
- [35] <http://www.globus.org/security/v1.1/openssh/install.html>, 2001.
- [36] <http://lexus.physics.indiana.edu/~griphyn/gripe/gripe.html>, 2001.
- [37] <ftp://achilles.ctd.anl.gov/pub/DEE/gsiklog-0.0.tar>, 2001.
- [38] Internet Engineering Task Force. <http://www.ietf.org>, 2001.
- [39] NASA Information Power Grid. <http://www.ipg.nasa.gov>, 2001.
- [40] W. Johnson, NASA's Information Power Grid: Implementation Experience with Production, Distributed Computing and Data Management Grids, Presented at the UK Office of Science and Technology's e-Science Town Meeting, QEII Conference Centre, London, <http://www.itg.lbl.gov/wej/Grids/#implementation>, July 27, 2001.
- [41] A.A. Khokhar, C.-L. Wang, M. Shaaban and V. Prasanna, Heterogeneous computing: Challenges and opportunities, *IEEE Computer Magazine, Special Issue on Heterogeneous Processing* **26**(6) (June 1993).
- [42] Legion, <http://www.cs.virginia.edu/~legion/>, 2001.
- [43] McCubbin, NORMAN, International ATLAS Overview, Atlas PCAP review discussion, available from http://heplocal.rl.ac.uk/zeus/ATLASvonR_30Oct01.ppt, October 2001.
- [44] MDS 2.1 Features in the Globus Toolkit 2.0 Release, <http://www-fp.globus.org/gt2/mds2.1/>, 2001.
- [45] National computational science alliance, <http://www.ncsa.edu/>, 2001.
- [46] Alliance science portals working group, <http://www.pablo.cs.uiuc.edu/scienceportals/index.htm>, 2001.
- [47] NEESgrid, National Virtual Laboratory for the Earthquake Engineering Science, <http://www.neesgrid.org>, 2001.
- [48] MICROSOFT, .net, <http://www.microsoft.com/net/>, 2001.
- [49] The Berkeley now project, <http://now.cs.berkeley.edu/>, 2001.
- [50] A national partnership for an advanced computational infrastructure, <http://www.npaci.edu/>, 2001.
- [51] NPI, New productivity initiative, <http://www.newproductivity.org>, 2001.
- [52] 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, also available from <http://www.globus.org/research/papers/myproxy.pdf>, August 2001.
- [53] C. Pancake, Can Users Play an Effective Role in Parallel Tools Research? Published in *International Journal of Supercomputing and HPC* **11**(1) (1997), 84–94, also available from http://www.cs.orst.edu/~pancake/papers/eff_role/.
- [54] Working group on peer-to-peer computing, <http://www.peer-to-peerwg.org/>, 2001.
- [55] J.M. Schopf and F. Berman, Stochastic scheduling, in *Proceedings of SuperComputing '99*, also available as Northwestern University, Computer Science Department Technical Report CS-99-3, or <http://www.cs.nwu.edu/~jms/Pubs/TechReports/sched.ps>, 1999.
- [56] The SC'xy conference series, SuperComputing: The International Conference for High Performance Computing and Communications, <http://www.supercomp.org/>.
- [57] J. Schopf, Grids: The Top Ten Questions, Seminar Talk, PDP/IT Seminar Series, CERN, Geneva, Switzerland, <http://www.cs.northwestern.edu/~jms/Pubs/TopTenTalk.pdf>, June 2001.
- [58] Seti@home, <http://www.seti-inst.edu/science/setiathome.html>, 2001.
- [59] J. Schopf, I. Foster, G. Fox, C. Pancake and M. Snir, Computational Grids: A solution looking for a problem?, Panel at SC '00, J Schopf Moderator, November 2000.
- [60] W. Smith, A Framework for Control and Observation in Distributed Environments, NASA Ames Technical Report NAS-01-006, June 2001.
- [61] J. Schopf and B. Nitzberg, Grids: The Top Ten Questions, Northwestern University, CS Dept. TR #CS-00-05, also available from <http://www.cs.northwestern.edu/~jms/Pubs/TopTen.pdf>, 2000.
- [62] B. Tierney, R. Aydt, D. Gunter, W. Smith, V. Taylor, R. Wolfski and M. Swany, A Grid Monitoring Architecture, GWD-Perf-16-1, also available at <http://www-didc.lbl.gov/GGF-PERF/GMA-WG/papers/GWD-GP-16-1.pdf>, 2001.
- [63] S. Tuecke, Grid Security Infrastructure Roadmap, GSI Working Group Grid Forum Draft, also available at http://www.gridforum.org/security/gf5_2000-10/drafts/draft-gridforum-gsi-roadmap-01.pdf, October 2000.

- [64] A. Tannenbaum, R. Van Renesse, H. Van Staveren, G. Sharp, S. Mullender and G. Van Rossum, Experiences with the amoeba distributed operating system, *Communications of the ACM (CACM)* **33**(12) (December 1990).
- [65] A.A. Ward, Trix: A network-oriented operating system, in *Proceedings of COMPCON*, 1980.
- [66] G. Wilson and H. Bal, An empirical assessment of the usability of orca using the cowichan problems, *IEEE Parallel and Distributed Technology* **4** (1996).
- [67] V. Welch, *Globus Security Architect*, Personal Communication, 2001.
- [68] R. Wolski, N. Spring and C. Peterson, Implementing a performance forecasting system for metacomputing: The network weather service, in *SC97*, November 1997.



Hindawi

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

