NORTHWESTERN UNIVERSITY

Computer Science Department

Technical Report
NWU-CS-00-5
2000

Grids: The top ten questions

Jennifer M. Schopf **Bill Nitzberg**

Abstract

The design and implementation of a national computing systems Grid has become a reachable goal for both computer and computational scientists. A distributed infrastructure capable of sophisticated computational functions can bring many benefits to scientific work, but poses many challenges to computer scientists, both technical and socio-political. Technical challenges include having basic software tools, 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.

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.

Grids: The Top Ten Questions

Jennifer M. Schopf*
Computer Science Department
Northwestern University
jms@cs.nwu.edu

Bill Nitzberg
PBS Products Department
Veridian Systems
bill@computer.org

Abstract

The design and implementation of a national computing systems Grid has become a reachable goal for both computer and computational scientists. A distributed infrastructure capable of sophisticated computational functions can bring many benefits to scientific work, but poses many challenges to computer scientists, both technical and socio-political. Technical challenges include having basic software tools, 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.

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, parallel distributed computing

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" [FRT78, Don79, War80]. In the late eighties and early nineties, these returned as distributed operating systems [Che88, TvRvS$^+$90, DO91].

*Corresponding Author

Shortly thereafter the field of heterogeneous computing [KWSP93, FS93] came to play: thousands of heterogeneous resources running hundreds of tasks. The next derivative of this area was parallel distributed computing in which parallel codes ran on distributed resources. This became metacomputing, and then computing on the Computational Grid [FK99].

There are a few 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 personal control over its resources, who gets accounts, usage policies, etc. Grids focus on 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 can be used. And the most important but perhaps the most subtle difference is the focus on the user. Previous systems were developed for and by the resource owner in order to maximize utilization and throughput. The Grid, however, focuses on the user. The specific machines that are used in a Grid 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.

A Grid is defined as a distributed infrastructure capable of sophisticated computational functions. Many groups are working on various pieces of Grid technology, as detailed in the following sections. In addition, members of the community have recently created the Grid Forum [Gri00] 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 and APIs.

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, 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.

1 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 they should be pervasive, dependable, consistent and inexpensive [FK99].

By combining these four properties Grids can have a transforming affect on how computation is performed and used.

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 solve a single application. Most often Grids are used for demos, supercomputing conferences, or other special occasions, and even in these cases extensive coordination between sites is required. This is primarily because Grids do not yet have the basic functionality needed for more extensive use.

So why hasn't this functionality been implemented? The answer is many-fold, and involves both technical and socio-political problems. One technical reason is because various groups assume someone else is working on a given piece of functionality, and so it is assumed that that group will solve it and others will not need to concern themselves with it. Alternatively, researchers may think such a basic concept has already been solved, but it has not been. In addition, it can be difficult to get funding to address basic functionality issues, as addressed in Section 10.

One example of the lack of functionality is the sharing resource information on the Grid. Groups are currently working on different information services [FFK+97, MDS00, GIS], however, there still needs to be an agreement on what the various terms mean. For example, when a monitoring service says "the load on a machine is X", a scheduling service must be able to interpret that information. What is needed is not a unilateral declaration of terms, but the agreement on standard usage that can be translated into local terms if need be. We see the advantage of this in common situations such as when a distance is reported to an American in meters. Most Americans don't use meters, but given a measurement in meters, they can translate it into a more familiar unit.

Work in basic functionality can be complicated by researchers being unclear under whose domain this work falls. In the case of standardizing some terms in the Grid Forum, it was originally assumed by the Performance and Scheduling Working Groups that the Information Working Group would coordinate this effort, while the Information Working Group assumed that others would see to it.

Below we address in more depth other basic functionality needed including variance management(see Section 6), cost models(see Section 7) and security(see Section 4).

Getting a Grid to function correctly is 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. In our information example above, not only does an information service need to be aware of the standard, but the monitors and whatever is using the monitors needs to be in agreement. This cooperation between research groups in different areas can be difficult.

In addition, because the various pieces of software or tools all 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).

Another reason why this work has been slow to develop is the lack of a testbed of machines that computer scientists can user to test the technologies they are developing. In order to adequately design and test basic functionality, computer scientists often need to work in an isolated or dedicated environment that they have complete control over. The Beta Grid Project [Bet00, Fos99] has begun to address this problem by setting up a national testbed for computer scientists to use to investigate some of the basic functionality approaches. However, this project is still in the planning stages [BCF00].

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

2 Why aren't here 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. Arguments 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 [BM98]). 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 [App00] as part of the Grid Forum. However, at the latest meeting of the Grid Forum (March 2000) that group realized that while there is a need for user input to guide the research, those users are hard to find [Ber00].

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 [NCS00], NPACI [NPA00], IPG [IPG00], GUSTO [GUS00], Condor [Con00], 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.

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 3). Standards must be developed to supply a uniform interface to Grid services (see Section 5). And basic functionality, as discussed in the last section, must become the default, not the exception.

Moral: *Until developers are given a better set of basic services, Grids will not be used to their full potential.*

3 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. 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 [Glo00, FK98], Legion [Leg00, GFKH99], and portals [Por00, Com00], among 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 such a project (see 10). For these tools to be more widely accepted, this lack of priority must change.

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

4 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 [ACP95, Now00]. And yet, for many systems, the mandatory "security is important" paragraph is all the depth the problem receives. In addition, for the work that has been done to address this (including [Leg00] and Globus [BEF$^+$99, FKTT98]), without agreement from all the sites on the Grid, it will not function. The need for standardization is one of the issues being addressed by the the Grid Forum Security Working Group [Sec00]. Standardization is addressed in more detail in Section 5.

Moral: *Without a security infrastructure users will not take advantage of the Grid.*

5 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 single answer to any of these questions.

One simple instance of this is 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*.

Some efforts have been made in this direction. Standardization is one of the goals of the Grid Forum in general. But since standardization isn't considered a "sexy" project by funding agencies, often there is little money, time or motivation to further this effort.

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

6 How can we manage variance on the Grid?

One of the primary difficulties in using the Grid is the unpredictable behavior of the resources. 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 [WSP97]. 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 [SB99] 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 should 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.*

7 What cost models are needed by the Grid?

Arguments have been made that the Grid will benefit by becoming an economic entity, where resource decisions can be made using all of the economic infrastructure that exists in a standard market economy [BAG00, CC00, Acc]. This implies having a form of currency that corresponds to cycles available to a user on a particular resource or at a specific site. Resource owners would be able to set the price of using a given resource individually. For example, the time on a popular, new machine might be more expensive than an older, less used resource. It is thought that in this way administrators would be able to manage utilization of a set of resources by adjusting their price.

However, it is very difficult to have a discussion of the market economy of the Grid when there is yet to be a sense of *cost*. Most research needing cost has simply ignored a detailed inspection of the concept or assumed it could be supplied by some oracle agent. In addition, for cost to become a usable concept, there needs to be some basic cooperation between political entities. Much as exchange rates are negotiated, the Grid will need negotiation agents to set the exchange rates between sites or resources. In this way, when a user requests resources, he or she can simply use cycle dollars for that expenditure. This would be one step closer to being able to get generic cycles from the Grid, however we are still a long way from achieving this.

Moral: *Cost is a concept that the Grid needs to address.*

8 Where are the Grid system management tools?

(This section is not included due to submission length constraints.)

Moral: *In order for the Grid to have the largest number of resources, administration of them must be simple and fast.*

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 Grids are centered around the needs of the users, as opposed to the needs of the resource owner. One of the driving motivations to creating the Grid was to offer users a way to take advantage of multiple resources, even when the managers of those resources could not agree either on technical or socio-political issues. However, sever problems arise because of this user-centric philosophy.

Since Grids are being developed to give benefits to individuals users, there isn't much direct advantage to the resource owners. The argument can, and has been made that the resource owners are there to help the users, but that is still an indirect benefit. This has the complication that without a direct benefit, many resource managers have seemingly been reluctant to install and deal with the infrastructure needed to take advantage of the Grid. This is one reason that the basic functionality has been slow to develop (see Section 1 above).

Another affect of this user-centric approach is that it puts the responsibility on the user to "play nicely". Sharing is hard—remember kindergarten? Or, for a more concrete example, think about the tragedy of the commons—individuals acting in their own best interests tend to overuse and degrade the resource. What benefit can the Grid offer to users that will keep them from being selfish?

Moral: *Benefits must be derived to curb the selfishness of both users and owners.*

10 How can we fund the research and development needed for the Grid?

As we've been stating, Grids need to have users to be successful. To have users, Grids must have basic functionality and standards. However, to do any work on the Grid, researchers need funding. Funding agencies don't seem to think basic functionality and standards are cool, so these get little or no funding support. Therefore, basic functionality and standards are ignored, thereby preventing more users from taking advantage of the Grid.

This is a problem being seen more and more clearly lately. Funding agencies are being put under a lot of pressure to fund only new and innovative projects. 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.

As another example, one of the best possible approaches that could be used in Grid research is to fund three to five projects to work on exactly the same problem. 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 five groups to do one thing is viewed as a waste of resources so this is unlikely to occur.

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.*

Conclusions

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 an SCXY demo can be run any time of the year.

Acknowledgements

We gratefully acknowledge the many folks who have had rambling conversations at meetings, universities, mailing lists, and bars which fostered this paper. We would especially like to thank Alain Roy, Ian Foster, Steve Tuecke, the Grid Forum Scheduling Working Group, and all the AppLeS folks. This work was support in part by NSF and the Alliance.

References

[Acc] Grid Forum Account Management Working Group. `http://www.nas.nasa.gov/~bthigpen/accounts-wg`.

[ACP95] Thomas E. Anderson, David E. Culler, and David A. Patterson. A case for networks of workstations: NOW. *IEEE Micro*, February 1995.

[App00] Grid Forum Application Working Group. `http://dast.nlanr.net/ GridForum/Apps-WG/`, 2000.

[BAG00] Rajkumar Buyya, David Abramson, and Jonathan Giddy. An economy driven resource management architecture for global computational power grids. In *Proceedings of the 2000 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2000)*, 2000. Also available from `http://www.dgs.monash.edu.au/~rajkumar/papers/GridEconomy.pdf`.

[BCF00] R. Butler, C. Catlett, and I. Foster. The Betagrid: Strawman Architecture. `http://dsl.cs.uchicago.edu/beta/bgstrawman.pdf`, 2000.

[BEF+99] R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, and V. Welch. Design and deployment of a national-scale authentication infrastructure. *Submitted*, 1999. Also available from `http://www-fp.globus.org/documentation/incoming/gsi-deploy.pdf`.

[Ber00] Francine Berman. Personal Communication, 2000.

[Bet00] Beta grid project. `http://dsl.cs.uchicago.edu/beta/`, 2000.

[BM98] M. Beck and R. Moore. The internet2 distributed storage infrastructure project. *Computer Networking and ISDN Systems*, pages 2141–2148, 1998.

[CC00] Brent N. Chun and David E. Culler. Rexec: A decentralized, secure remote execution environment for clusters. In *To appear in 4th Workshop on Communication, Architecture, and Applications for Network-based Parallel Computing*, 2000. Also available from `http://www.cs.berkeley.edu/~bnc/papers/canpc00.pdf`.

[Che88] David Cheriton. The V distributed system. *Communications of the ACM (CACM)*, 31(3), March 1988.

[Com00] Computing portals. `http://www.computingportals.org/`, 2000.

[Con00] Condor: High throughput computing. `http://www.cs.wisc.edu/condor/`, 2000.

[DO91] Fred Douglis and John Ousterhout. Transparent process migration: Design alternatives and the sprite implementation. *Journal of Software–Practice & Experience*, 21(8), August 1991.

[Don79] W. J. Donnelley. Components of a network operating system. *Computer Networks*, 3(6):389–399, December 1979.

[FFK+97] S. Fitzgerald, I. Foster, C. Kesselman, G. von Laszewski, W. Smith, and S. Tuecke. A directory service for configuring high-performance distributed computations. In *Proceedings of the 6th IEEE Symp. on High-Performance Distributed Computing*, 1997. Also available from `ftp://ftp.globus.org/pub/globus/papers/hpdc97-mds.pdf`.

[FK98] Ian Foster and Carl Kesselman. The Globus project: A status report. In *Proceedings of IPPS/SPDP '98 Heterogeneous Computing Workshop*, 1998. Also available from `ftp://ftp.globus.org/pub/globus/papers/globus-hcw98.pdf`.

[FK99] Ian Foster and Carl Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, Inc., 1999.

[FKTT98] 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*, 1998. Also available from `ftp://ftp.globus.org/pub/globus/papers/security.pdf`.

[Fos99] Ian Foster. The Betagrid: A national infrastructure for computational systems research. In *Proceedings of NetStore '99*, 1999.

[FRT78] H.C. Forsdick, R.E.Schantz, and R. H. Thomas. Operating systems for computer networks. *Computer*, 11(1):48–57, January 1978.

[FS93] R.F. Freund and H.J. Siegel. Heterogeneous processing. *IEEE Computer*, pages 13–17, June 1993.

[GFKH99] Andrew Grimshaw, Adam Ferrari, Fritz Knabe, and Marty Humphrey. Legion: An operating system for wide-area computing. Technical Report CS-99-12, University of Virginia, Computer Science Dept., March 1999. Also available from `ftp://ftp.cs.virginia.edu/pub/techreports/CS-99-12.ps.Z`.

[GIS] Grid Forum Information Services Working Group. `http://www-unix.mcs.anl.gov/gridforum/gis/`.

[Glo00] Globus. `www.globus.org`, 2000.

[Gri00] Grid Forum. `http://www.girdforum.org`, 2000.

[GUS00] Globus Ubiquitous Supercomputing Testbed. `http://www-fp.globus.org/overview/testbeds.html`, 2000.

[IPG00] NASA Information Power Grid. `http://www.nas.nasa.gov/About/IPG/ipg.html`, 2000.

[KWSP93] Ashfaq A. Khokhar, Cho-Li Wang, Mohammad Shaaban, and Viktor Prasanna. Heterogeneous computing: Challenges and opportunities. *IEEE Computer Magazine, Special Issue on Heterogeneous Processing*, 26(6), June 1993.

[Leg00] Legion. `http://www.cs.virginia.edu/~legion/`, 2000.

[MDS00] Globus Metacomputing Directory Service. `http://www-fp.globus.org/mds/`, 2000.

[NCS00] National Computational Science Alliance. `http://www.ncsa.edu/`, 2000.

[Now00] The Berkeley NOW project. `http://now.cs.berkeley.edu/`, 2000.

[NPA00] A National Partnership for an Advanced Computational Infrastructure. `http://www.npaci.edu/`, 2000.

[Por00] Alliance Science Portals Working Group. `http://www-pablo.cs.uiuc.edu/scienceportals/index.htm`, 2000.

[SB99] Jennifer M. Schopf and Francine Berman. Stochastic scheduling. In *Proceedings of SuperComputing '99*, 1999. Also available as Northwestern University, Computer Science Department Technical Report CS-99-3, or `http://www.cs.nwu.edu/~jms/Pubs/TechReports/sched.ps`.

[Sec00] Grid Forum Security Working Group. `http://www.ncsa.uiuc.edu/General/GridForum/SWG`, 2000.

[TvRvS+90] 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.

[War80] A. A. Ward. Trix: A network-oriented operating system. In *Proceedings of COMPCON*, 1980.

[WSP97] R. Wolski, N. Spring, and C. Peterson. Implementing a performance forecasting system for metacomputing: The network weather service. In *SC97*, November 1997.