

The Grid: Feng Shui for the Terminally Rectilinear

Martha Stewart

Introduction

While the rapid evolution of “The” Internet continues to define a new medium for the sharing and management of information, it also brings with it the potential for harnessing vast numbers of computers, storage devices, and networks as a platform for computation. Developing and running programs that can draw compute power from globally distributed resources poses new challenges for the computer and computational science communities. In addition to interoperability and security (which are key concerns for Internet users and developers as well), applications that use distributed resources as a unified compute platform must be able to achieve performance levels that are greater than any single resource could deliver alone.

The *Computational Grid* [13] is a software approach, the main goal of which is to aggregate resources that are culled from a global resource pool for use by applications and their users. Using a ubiquitous software veneer that provides a consistent set of services, Grid applications should be able to draw compute power from a federated collection of resources (i.e. generators) much in the same way that electrical appliances draw electrical power from a power utility. Building the software infrastructure that is capable of realizing this ambitious metaphor is the focus of many research and development efforts at present. This article attempts to survey “the state” of Computational Grid computing in terms of some, but by no means all, of the currently active Grid efforts. We’ve categorized on-going work in this area according to *current practice*, *research efforts*, and *commercial developments* but we stress that the projects we’ve chosen to describe (particularly the research projects) are intended only to be a representative sampling from a population too large to detail exhaustively.

Current Practice

Grid software infrastructures that implement basic services have matured to the point where they have engendered active user communities. They are portable between all of the currently available Unix and Linux implementations, and are able to run on all of the architecture types used for scientific and engineering computing (e.g. parallel supercomputers, workstations, parallel shared-memory machines, laptops, etc.) Compatibility with the Microsoft operating systems is increasingly becoming a requirement,

and development is underway, but as of yet, little in the way of “production quality” software is available for these platforms.

The Grid software infrastructure in widest use today is Globus [15] – a software toolkit for building Grid-enabled applications and services. Globus implements a set of non-proprietary protocols for securely identifying, allocating, and releasing resources within a Grid (i.e. from a globally federated pool). Applications use directly a variety of programming tools and libraries that implement these protocols to locate and gain access to Grid resources. At the same time, any operation can be authenticated via the Globus Grid Security Infrastructure (GSI) [21] mechanisms (using X.509 certificates, by default) which are built into all software components. Globus is available as open source, portable amongst all versions of Unix and Linux (including those that run on supercomputers, shared-memory parallel machines, clusters, etc.), and is used by a large number of the active Grid projects today.

Another mature and widely used infrastructure is Condor [10]. Originally developed as a “cycle-scavenging” system for workstation networks, Condor’s developers have adapted it to work in wide-area settings, and have added parallel supercomputers, clusters, shared-memory multiprocessors as target machine types. Condor attempts to maximize job throughput by borrowing time from idle resources. Resource owners who agree to run Condor specify the conditions under which Condor may acquire and must release their resources. When a resource becomes eligible for acquisition, Condor is free to schedule jobs on it. If processor or keyboard activity indicates that the resource is no longer idle, Condor checkpoints any jobs executing on the resource and evacuates them.

Condor also supports a high-level resource discovery mechanism called *matchmaking* that allows users to specify their resource needs in a high-level language. When the job is launched, Matchmaker service performs a best-fit search of the resource database to determine the resources that should be allocated.

Globus and Condor interoperate. Condor-G (as the combination is called) uses a combination of the Condor resource allocation strategies (including matchmaking) and the Globus security and resource access mechanisms. The result is a secure, portable, and efficient system for implementing a high job-throughput capability in Grid settings.

Grid Projects

There are several large-scale Grid computing efforts underway, most if not all of which use Globus, Condor, or Condor-G in some capacity, but which also include research products from the wider community. The TeraGrid [38] Project, funded by the National Science Foundation, is deploying a national-scale multi-institution Grid to support scientific computing with an expected peak performance of 13.5 teraflops. TeraGrid users will be able to log-on to a nationally distributed collection of high-performance resources and to treat them as a single heterogeneous computational and storage platform. Other Grid development and deployment projects have similar goals and scope, but typically with an emphasis on a particular application domain or class of problems. These projects include the Grid Physics Network (GriPhyN) [20], the Network for Earthquake Engineering System (NEES) [27], the NASA Information Power Grid [24], the Department of Energy’s Particle Physics Data Grid (PPDG) [33], the European Data

Grid [12], and the Asia-Pacific Grid [2]. While the state of development and size of the user community supported by these projects varies, taken together they constitute a cross-section of effective Grid planning and implementation today.

Research Efforts

Research focusing on how to build and use Computational Grids continues to expand rapidly, yielding a plethora of experimental execution environments, schedulers, modeling systems, and user-interface tools. Indeed, so much work is underway that an exhaustive enumeration of these systems and their results at present is difficult to imagine¹. Instead, our intention is to survey representative research efforts as an entree to the wider field.

Perhaps the most comprehensive single Grid research project, in terms of its scope, is the Grid Application Software (GrADS) [17] Project centered at HiPerSoft [22]. GrADS investigates an integrated approach to Grid program development and execution that includes automatic Grid-enabled libraries, compilation and compile-time optimization techniques targeting the Grid, high-performance Grid runtime scheduling systems, dynamic application monitoring and control, new Grid simulation capabilities, and market-based Grid resource allocation strategies. While these foci, taken individually, are each an active area of research, GrADS is studying what is necessary to combine them into an effective programming and execution environment for the user.

The GrADS results generated in these research areas build upon those from a number of seminal and currently active Grid projects, including AppLeS [3] for application scheduling, Autopilot [5] for application monitoring and control, MPICH-G [26] for MPI support, The Network Weather Service [31] for resource monitoring and performance forecasting, and ScaLAPACK [35] for numerical libraries and distributed performance tuning. GrADSoft (the experimental software environment developed by GrADS) should also be able to leverage other Grid programming systems such as NetSolve [28] and Ninf [30] that provide Remote Procedure Call (RPC) services that are Grid “aware.”

By itself, the subproblem of storage management in Grid environments is the subject of active research. The Storage Resource Broker (SRB) [37] project is investigating infrastructure requirements that must be met to provide a uniform interface to storage across heterogeneous resources. Using a different model for distributed storage, the Internet Backplane Protocol (IBP) [23] project considers the use of storage depots from which clients can make temporary allocation requests anonymously. While their approaches are different and complementary, both of these projects are focused on optimizing data deliver to Grid client applications from disparate storage sites.

Complementing the computational and storage capabilities investigated by projects such as GrADS and its related work, the AccessGrid [1] Project studies how the Grid can be used to foster human interaction and collaboration. In addition to “standard” video conferencing capabilities, the AccessGrid is designed to support collaborating

¹A web search using www.google.com for the key words “computational grid” on 12/11/01 yielded 743 relevant web pages.

teams of users who require more than simple audio and visual connectivity to interact. Users can participate in “virtual venues” that are built using Multi-User Domain (MUD) environments and multi-media systems.

Providing a Grid computing capability via the World-wide Web is also an active research area. The Purdue University Network Computing Hubs (PUNCH) Project [34] uses a set of runtime resource brokering services to identify potential execution sites for a user’s job. It then launches the job and provides a web-browser interface for its status and control.

PUNCH is an example of a wider effort to provide web-portal like access to Grids [19]. In a similar vein, the Indiana University Common Component Architecture Toolkit (CCAT) Project [8] is studying ways that Grid software component interfaces can be standardized and maintained. The Globus CoG Kit [9] extends this notion to include technologies and tools that have been developed originally for Internet applications and web services.

In addition, researchers are developing application frameworks for specific types of applications that share common characteristics. The Cactus development environment [7] provides an object-oriented development framework that has been used, primarily, to implement large-scale physics applications for the Grid. Simulation parameter “sweeps” in which an individual simulation is to be run repeatedly using a different parameterization each time is also a successful class of Grid application. Both the Nimrod [29] and AppLeS Parameter Sweep Template (APST) projects [4] are studying ways to optimize the execution of these scientific applications.

Commercial Endeavors

The success of Grid research, and the user community it has attracted, has generated commercial interest in the Computational Grid paradigm. Recently, Compaq, Cray, Silicon Graphics, Sun Microsystems, Veridian, Fujitsu, Hitachi, NEC, Entropia, IBM, Microsoft, and Platform Computing announced plans to develop commercial offerings based on Globus [16]. These companies intend to leverage the open standard that Globus has fostered in the community as the basis for new Grid products.

Spin-off companies are also flourishing. Sun Microsystem’s Grid Engine product line [18] stems from an early company acquisition. The Avaki Corporation [6] is offering a commercialized version of Legion [25] — a fully object-oriented Grid system originally developed at the University of Virginia.

Other companies are exploring peer-to-peer Internet technologies as a way of providing commercial Grid computing options. Entropia [11], United Devices [39], and Paragon [32] offer distributed systems for enterprise wide or Internet-based Grid computing. Users “sign-up” by activating a client library that allows both job submission and safe cycle harvesting. In addition, users from the World-wide Web are encouraged to participate in public computing projects much in the same way that the Search for Extraterrestrial Intelligence (SETI) Project recruits volunteers for SETI@Home [36].

Conclusion

The Computational Grid is an emerging distributed computing paradigm with active research, user, and commercial development communities. Relatively mature software infrastructures that are freely available have engendered a host of large-scale development and deployment efforts. Experimental research that investigates more powerful and easier-to-use techniques continues, and commercial interest is robust.

With so much interest in the future of Computational Grids, The Global Grid Forum (GGF) [14] has been formed as a community-based organization for information interchange and standardization. Modeled after the Internet Engineering Task Force (IETF), GGF participants (organized as working groups) address issues such as security, uniform access to information services, data management, performance monitoring, and application models as a community. As a body, the GGF meets three times a year internationally, and wide participation is welcome.

The networking revolution that launched The Internet continues to open new possibilities with the advent of the Computational Grid. If successful, computing power will be available from a utility much in the same way that electrical power, cable T.V., and Internet service are today.

References

- [1] The AccessGrid Home Page. <http://www-fp.mcs.anl.gov/fl/accessgrid>.
- [2] The Asia Pacific Grid Home Page. <http://www.apgrid.org>.
- [3] The AppLeS Home Page. <http://apples.ucsd.edu>.
- [4] The AppLeS Parameter Sweep Template Home Page. <http://grail.sdsc.edu/projects/apst>.
- [5] The Autopilot Home Page. <http://www-pablo.cs.uiuc.edu/Software/Autopilot/autopilot.htm>.
- [6] The Avaki Home Page. <http://www.avaki.com>.
- [7] The Cactus Code Home Page. <http://www.cactuscode.org>.
- [8] The Common Component Architecture Toolkit Home Page. <http://www.extreme.indiana.edu/ccat>.
- [9] The Globus CoG Home Page. <http://www.globus.org/cog>.
- [10] The Condor Home Page. <http://www.cs.wisc.edu/condor>.
- [11] The Entropia Home Page. <http://www.entropia.com>.
- [12] The Data Grid Home Page. <http://www.eu-datagrid.org>.
- [13] I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, Inc., 1998.
- [14] The Global Grid Forum Home Page. <http://www.gridforum.org>.
- [15] Globus. <http://www.globus.org>.
- [16] 12 Companies Adopt Argonne Lab/USC Globus Toolkit as Standard Grid Technology Platform. <http://www.globus.org/developer/news/20011112a.html>.
- [17] The GrADS Home Page. <http://hipersoft.cs.rice.edu/grads>.
- [18] The Grid Engine Home Page. <http://www.sun.com/gridware>.
- [19] The Grid Portal Collaboration Home Page. <http://www.ipg.nasa.gov/portals>.
- [20] The Grid Physics Network Home Page. <http://www.gridphyn.org>.
- [21] The Grid Security Infrastructure Home Page. <http://www.globus.org/security>.

- [22] The HiPerSoft Home Page. <http://www.hipersoft.rice.edu>.
- [23] The Internet Backplane Protocol Home Page. <http://icl.cs.utk.edu/ibp>.
- [24] The NASA Information Power Grid Home Page. <http://www.ipg.nasa.gov>.
- [25] The Legion Home Page. <http://legion.virginia.edu>.
- [26] The MPICH-G Home Page. <http://www.niu.edu/mpi>.
- [27] The Network for Earthquake Engineering and Simulation. <http://www.neesgrid.org>.
- [28] The NetSolve Home Page. <http://icl.cs.utk.edu/netsolve>.
- [29] The Nimrod Home Page. <http://www.csse.monash.edu.au/~david/nimrod.html>.
- [30] The Ninf Home Page. <http://ninf.apgrid.org>.
- [31] The Network Weather Service Home Page. <http://nws.npaci.edu>.
- [32] The Parabon Home Page. <http://www.parabon.com>.
- [33] The Particle Physics Data Grid Home Page. <http://www.slac.stanford.edu/xorg/ngi/ppdg/ppdg-slac.html>.
- [34] The Purdue University Network Computing Hubs Home Page. <http://punch.ecn.purdue.edu>.
- [35] The ScaLAPACK Home Page. <http://www.netlib.org/scalapack>.
- [36] The SETI@Home Web Page. <http://setiathome.ssl.berkeley.edu>.
- [37] The Storage Resource Broker Home Page. <http://www.npaci.edu/DICE/SRB>.
- [38] The TeraGrid Home Page. <http://www.teragrid.org>.
- [39] The United Devices Home Page. <http://www.ud.com/home.htm>.