Building the TeraGrid

Peter H. Beckman[1]
University of Chicago/Argonne National Laboratory, Argonne IL, 60439, USA

Summary

On October 1, 2004, the most ambitious high-performance Grid project in the United States, the TeraGrid, became fully operational. Resources at nine sites—the San Diego Supercomputer Center, the California Institute of Technology, the National Center for Supercomputing Applications, the University of Chicago/Argonne National Laboratory, Pittsburgh Supercomputer Center, Texas Advanced Computing Center, Purdue University, Indiana University, and Oak Ridge National Laboratory—were joined via an ultra-fast optical network, unified policies and security procedures, and a sophisticated distributed computing software environment. Funded by the National Science Foundation, the TeraGrid enables scientists and engineers to combine distributed, multiple data sources with computation at any of the sites or link massively parallel computer simulations to extreme-resolution visualizations at remote sites. A single shared utility lets multiple resources be easily leveraged and provides improved access to advanced computational capabilities. One of the demonstrations of this new model for using distributed resources, TeraGyroid, linked the infrastructure of the TeraGrid with computing resources in the United Kingdom via a trans-Atlantic data fiber link. Once so connected, the software framework of the RealityGrid project was used to successfully explore lattice-Boltzmann simulations involving lattices of over one billion sites.

Introduction

Applications from the science and engineering community are becoming increasingly sophisticated and complex. Multidisciplinary teams work with computer scientists to integrate a wide range of components, such as databases, satellite images, simulation frameworks, visualization engines, and instrument data. The decades-old model of a single scientist building a monolithic code has given way to application frameworks built by dozens of scientists from several different fields. High-end simulations use multiscale, multicomponent models for which no one person can be an expert for all software components and data sets. Successful groups have combined their resources, both human and technical to focus on scientific discovery.

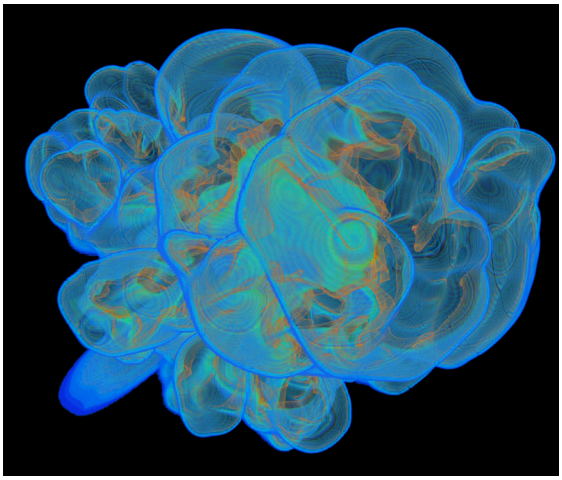

Figure 1: Volume rendering of an unstable flame front from a simulation of the deflagration phase of a Type Ia supernova; the University of Chicago TeraGrid visualization cluster provides advanced visualization capabilities to TeraGrid users. (Image courtesy of the University of Chicago Flash Center)

The purpose of the TeraGrid is to provide a unified distributed computing facility that can be the foundation of these new, complex, computational science activities. The construction project faced

[1] This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38, and in part by NSF under Grant No. ANI-01-22296.

significant challenges. Building a nationwide facility stretched the limits of the existing software, and several important components required custom solutions. Moreover, the social hurdles were possibly more challenging than the technical ones. Missing functionality can be identified and programmers assigned to fill the gap; however, the complex issues of control, security, funding, and name recognition are not so easily dispatched. The remainder of this article discusses Grid computing, the architecture and design of the TeraGrid, and some of the lessons learned during the construction of the facility.

Evolution of the TeraGrid

The genesis of the TeraGrid project was the Distributed Terascale Facility (DTF). The DTF was conceived in October 2001 as a distributed high-performance computing Grid that could link four supercomputers as well as some specialized resources for data management and visualization. The compute platforms were nearly identical Linux clusters built by IBM and based on Intel's IA64 Itanium architecture. A portion of the DTF was installed at each of the four partner sites: the San Diego Supercomputer Center (SDSC), the California Institute of Technology (CIT), the National Center for Supercomputing Applications (NCSA), and the University of Chicago/Argonne National Laboratory (UC/ANL). To provide the bandwidth required for high-end applications to use the four sites as a single utility, TeraGrid partnered with Qwest Communications for a transcontinental optical network providing an amazing 40 Gb/s between hubs in Chicago (CHI) and Los Angeles (LA). Each of the DTF sites is connected to the hubs at 30 Gb/s. To put that amazing bandwidth into perspective, the 40 Gb/s link between Chicago and Los Angeles, if used for voice telephone calls, would be capable of handling the traffic of over half a million simultaneous callers.

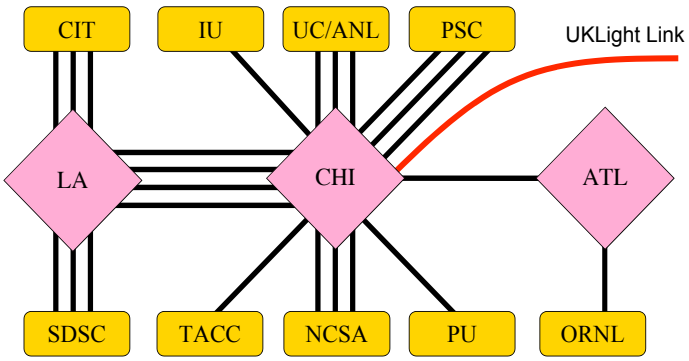

Figure 2: TeraGrid network of 10 Gb/s lambdas

In October 2002, the Pittsburgh Supercomputer Center (PSC) was added to the DTF, and the Extensible Terascale Facility (ETF) project was started. The HP Alpha-based supercomputers at PSC brought heterogeneity to the project and accelerated the planned move from environments that were binary compatible to one based on standardized software and services. In October 2003, the NSF added Indiana University (IU), Purdue University (PU), The Spallation Neutron Source at Oak Ridge National Laboratory (ORNL), and the Texas Advanced Computing Center (TACC). The network was also extended and a new hub in Atlanta (ATL) was created (see Figure 2). Collectively, the nine sites—known as the TeraGrid—bring together a remarkable and growing array of high-end facilities in support of science. Table 1 shows the resources available within the TeraGrid expected in 2005.

Building a High-Performance Production Computing Grid

Today, the word "Grid" has become a catchall term for what was once loosely called distributed computing. Hardware companies have renamed product lines, turning rack-mounted servers with clustering software into "compute Grid rack systems." Software companies sell "Grid Databases" and build on-line gaming systems for Sony PlayStations that exploit "Grid technology." In some scientific communities, Grid computing refers to CPU scavenging, where idle machines are converted into a shared computing resource, such as the system provided by SETI@home (Sullivan *et al.* 2001) to search for extraterrestrial life. Other communities believe Grid computing is made possible by a Web front-end to remote job submission, where an automated job broker chooses the resource to use for the job. To further complicate the playing field, there are two industry standards groups, the Global Grid Forum, which

started in 1999, and the Enterprise Grid Alliance (EGA), founded in 2004. Together, the two bodies host groups that span topics from "utility computing" (on-demand servers billed by the hour), to peer-to-peer computing and data replication.

For the TeraGrid, however, Grid computing refers to the software, policies, and procedures that enable the integration of high-end computational facilities into a unified system supporting advanced science and engineering research. One of the first, and possibly most important, lessons learned during the construction of the TeraGrid is that the word "Grid" must be replaced by extremely precise definitions. Will a shared global file system be provided? What, exactly, must scientists do to get an account, and where can it be used? Unambiguous details of both policies and construction plans are key both to matching the expectations of users and funding agencies and to planning and management. Naturally, since the views are so divergent, a Grid system cannot serve all users who identify themselves as needing a Grid computing system.

In addition to the scientific community and the raw computational resources, a high-performance production Grid infrastructure requires four components: a set of shared policies; a well-defined, maintained, and tested set of software and services; a "User Services" support team to work directly with scientists; and a distributed operations, networking, and security center. The design and construction of these four components depend heavily on the user community, applications, and funding model. The Open Science Grid and EGEE (Enabling Grids for E-Science and Europe) for example, are largely volunteer or best-effort Grids. Sites participate by offering up and joining together community resources. Other Grids, such as the TeraGrid, the National Grid Service (NGS) of the UK e-Science Programme (Allan *et al.* 2003), and DEISA (Distributed European Infrastructure for Supercomputing Applications) are focused on production resources with service level guarantees and a more centralized management of users, accounts, and security. In the sections below, the motivation, design, and implementation of the TeraGrid's four major components are presented.

	Compute Resources	**Online Storage**	**Archive Storage**	**Network & Hub**	**Data Collections**	**Instruments**	**Visualization**
UC/ANL	1 TF	20 TB		30 Gb/s to CHI			Yes
CIT	0.8 TF	155 TB		30 Gb/s to LA		Yes	
IU	2.2 TF	32 TB	1.2 PB	20 Gb/s to CHI	Yes	Yes	Yes
NCSA	16.5 TF	600 TB	3.0 PB	30 Gb/s to CHI	Yes		
ORNL		1 TB		10 Gb/s to ATL		Yes	
PSC	16.3 TF	150 TB	2.4 PB	30 Gb/s to CHI			Yes
PU	1.7 TF			10 Gb/s to CHI	Yes		Yes
SDSC	10.8 TF	540 TB	6 PB	30 Gb/s to LA	Yes		
TACC	6.3 TF	50 TB	2 PB	10 Gb/s to LA	Yes		Yes
Total	**56 TF**	**1.5 PB**	**15 PB**				

Table 1: TeraGrid Resources in 2005

Shared Policies

For the TeraGrid, focusing on the shared vision and goals early in the design and construction process helped guide the evolution of policies and governance. Below are the core shared policies for building the TeraGrid.

- *A single identity with multiple computer accounts:* Public key encryption provides the basis for Grid middleware that uses X.509 certificates for authentication. Users have simplified access to resources

by maintaining a single identity even though each computer center in the TeraGrid may map that identity to a different local Unix account. The TeraGrid is exploring a shared global file system, but no suitable heterogeneous solution currently meets the needs of the project.

- *A single accounting currency, the "TeraGrid Service Unit":* Scientists apply for TeraGrid resource allocations by submitting proposals that are peer reviewed by national resource allocation committees. Resource use (in terms of CPU-hours) is carefully tracked.
- *A unified support team:* With nine different computer centers participating in the TeraGrid, users must have a simple, coherent interface for interacting with the TeraGrid. The TeraGrid maintains a single account request form, a single email address where questions can be submitted, a single set of usage policies, and a single toll-free telephone help line for users. Users see a unified support team.
- *A unified infrastructure:* Resource virtualization is not practical with today's software and Grid middleware. An application source code generally must be compiled for each type of compute platform it can run. The TeraGrid strives to remove arbitrary differences that would require scientists to maintain special code for each platform or site. Where differences must remain, software layers to hide the differences are sought.
- *A unified documentation and training program:* Often, using multiple resources is hampered by the human investment required to learn the operation of each component of the system. Striving for a unified infrastructure, to smooth out the technical barriers to distributed computing, is an important first step. Second is a unified documentation and training program. The TeraGrid attempts to create a "Learn Once, Run Anywhere" model. Getting a TeraGrid account, allocation, and attending a tutorial or reading the documentation should be sufficient to start using any of the TeraGrid resources.

Coordinated TeraGrid Software and Services

The heart of any distributed computing system is the software that provides coordinated services to the scientists building applications. The first generation of services that evolved to span the Internet included Telnet, FTP, and eventually HTTP. In response to the need to eliminate clear-text passwords and support encrypted channels, Secure Shell (SSH) replaced Telnet in most environments. However, these services provide only the simplest form of remote access and file transfer capabilities. In order to build a rich set of distributed computing services, higher-level services are required. Grid middleware permits applications to use multiple, distributed resources easily. For example, many applications need to remotely access data; create, and manage multiple tasks; and use specialized, remote components for computation or visualization. To provide these capabilities, TeraGrid uses a coordinated set of Grid middleware, including the Globus Toolkit (Foster & Kesselman 1998), OpenSSH, SRB (Baru *et al.* 1998), MyProxy (Novotny, Tuecke & Welch 2001), and Condor-G (Frey *et al.* 2001). These and many other packages provide the user with an environment in which single sign-in, remote job submission, and access to distributed data is possible. A distributed accounting system, developed for the TeraGrid, permits each site to continue using their existing, often customized accounting system to collect usage data. The locally gathered data is converted into standard record formats and a centralized database collects project-wide user accounting data.

Providing this shared Grid-enabled infrastructure demands careful coordination, engineering, and planning. Every scientist understands the impossibility of building reliable complex systems using components that are not persistent, standard, or consistently configured throughout the system. TeraGrid addressed this issue by organizing TeraGrid software and services, the processes by which they are deployed, and the mechanisms by which they are tested and measured. The result—the *Coordinated TeraGrid Software and Services (CTSS)* —provides a robust, persistent, stable target for the application development and execution environments using the following strategy:

- Wherever possible, remove arbitrary differences in the key software and services. Otherwise, users are forced to encode that complexity in their application.
- Within capability classes (compute, storage, visualization, instruments, databases) synchronize versions and configurations of key software and service components where multiple, standards-based

implementations do not exist: for example, synchronized Grid middleware to provide a single version of the Globus Toolkit (identical patches, configurations) or a synchronized version of visualization libraries and run-time environments.

- Where multiple, standards-based implementations exist, ensure that users can exploit the capability without adding complexity to their code. For example, MPI and Fortran 90 is part of the CTSS-compute environment, but the specific, standards-based software packages do not have to be identical.

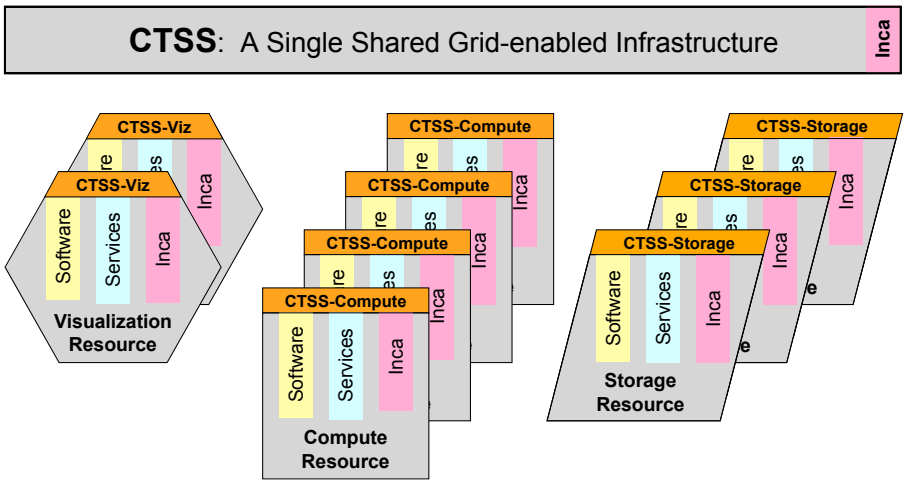

Figure 3: CTSS components provide a unified architecture

CTSS 2.0, in place today, addresses computation, storage, and visualization (see Figure 3). Each resource provider may provide one or more of the capability classes; however, the user sees a unified infrastructure. In most cases CTSS does not prevent a resource provider from providing multiple versions or configurations in addition to the tested CTSS environment (e.g., multiple versions of the Globus Toolkit, compilers, MPI, math libraries).

This approach is substantially more rigorous than that of most Grid environments and provides a more carefully defined set of components. CTSS comprises more than 40 different packages, combined with an advanced environment management system that maintains more than 75 synchronized software key names for packages, so precise versioning and use of tool sets are simplified across TeraGrid sites.

This careful coordination of building blocks was designed to improve the services for applications such as the National Virtual Observatory (Brunner *et al.* 2002), which is constructing data-mining pilelines for astronomical data, and the Linked Environments for Atmospheric Discovery (LEAD), a collaboration to build infrastructure capable of doing better than real-time forecasts of mesoscale weather events such as tornadoes (Xue *et al.* 2003). While this approach simplifies the environment for the user community, however, the tight coordination can complicate the efforts of the resource providers. Decisions concerning which software packages will be supported as part of CTSS can also be very difficult to manage across nine institutions serving communities with sometimes differing requirements.

Verification: Sophisticated Monitoring, Testing, and Quality-of-Service Metrics

Because Grid middleware did not provide the rigorous level of component and quality-of-service (QoS) tests required for production computing, the TeraGrid developed a monitoring, testing, and data collection infrastructure, called *Inca,* as an essential component of CTSS. Inca currently collects over 900 pieces of data about TeraGrid resources, providing a comprehensive snapshot of the system (see Figure 4). This snapshot helps users and administrators understand the software environments available to applications;

to measure overall availability of the Grid software and services within TeraGrid; and to identify, track, and diagnose resource outages from the *user* point of view, often undetected by traditional resource monitoring. The historical data collected by Inca also assists the quality assurance efforts of the TeraGrid. From the summary percentages tracked over time, sites can gauge their effectiveness in providing the core TeraGrid services.

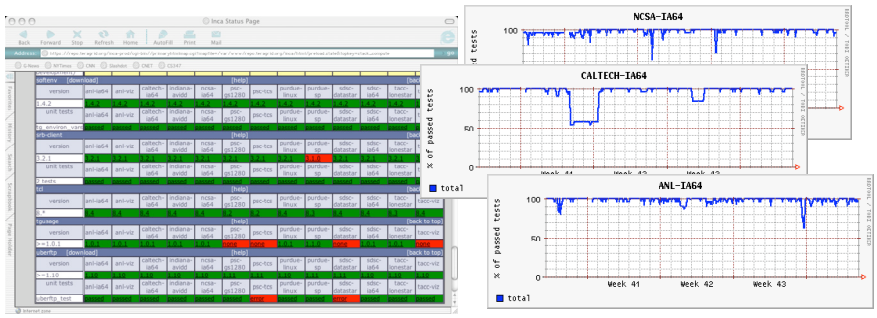

Figure 4: Inca provides detailed information status and historical performance of CTSS across the TeraGrid

User Services Support Team

As scientists expand and evolve application codes to make use of unified TeraGrid resources, a single support team that can provide documentation, training, help, performance tuning, and debugging assistance is vital. Experience has shown that application teams will migrate toward one or two well-supported platforms rather than interact with a handful of organizations and dissimilar systems, so a unified support team is required. The user services team is geographically distributed and responsible for initial contact with users and maintains and organizes the following:

- *A nationwide open allocation request process:* User Services helps the NSF organize the solicitation for computer resources, and works with the community to prepare submissions. Allocations may be requested for "roaming" TeraGrid Service Units, CPU cycles that may be spent on nearly any resource, as well requests for a particular resource.
- *Tutorials and training:* At both the Supercomputing 2003 and Supercomputing 2004 conferences, the User Services team presented tutorials on using the TeraGrid.
- *Help and new user assistance:* User Services works with both new and experienced users when they need assistance with their application. If the User Services team cannot answer the question, they draw on the knowledge of the TeraGrid's working groups that provide the technical support for networking, software, and data.
- *Performance tuning assistance:* The User Services team works with the performance evaluation working group to help users tune their applications for the TeraGrid.
- *Account management:* User Services handles both the creation of new accounts and the maintenance of the basic accounting system that tracks TeraGrid Service Units.

Experience gained during TeraGrid construction has shown that an active, responsive user services organization is key to adoption of new computing models and advanced software. Furthermore, for a successful Grid, users required a single, simple interface to the distributed user services team.

Operations, Networking, and Security

The TeraGrid constructed a 24x7 operations center for the TeraGrid by utilizing the existing resources at SDSC and NCSA. The two centers share a toll-free telephone line that shifts between the two sites every 12 hours and a trouble-ticket system that is shared among all TeraGrid partners. Staff are always available to answer questions via telephone or email. Questions that the operations center cannot answer

or solve promptly are dispatched to other working groups and examined during normal business hours. Specialized security and networking teams handle time-critical issues by using a call-tree.

One lesson learned during TeraGrid construction was that trouble tickets needed to be assigned to individuals, not geographically distributed working groups. Often, no one took personal responsibility for identifying the problem and resolving it. The ticket system and the set of procedures used to assign tickets was modified to assign tickets to individuals, as well as notify the site lead that a ticket had been assigned. This simple change helped improve service.

Security in a Grid Environment

Keeping computer systems secure requires continuous effort, especially for a research-driven infrastructure that is frequently updated with experimental Grid-enabled software. One factor that has made building a secure TeraGrid difficult is its visibility. The TeraGrid, as a whole, certainly ranks as one of the largest integrated computer systems available today. That visibility makes it an attractive target. The reason lies not in the technology but in the perceived value in compromising the security of the system—whether purely to increase one's reputation in the hacker community or to use the system for a massive denial-of-service attack. For the architects of the TeraGrid, this visibility translates directly into a nonlinear cost for cybersecurity. The larger the system, the larger the target, and therefore the larger the fraction of the enterprise that must be devoted to locking out intruders and detecting them once they have compromised the system. Breeches are inevitable, and resources must also be devoted to detection, response, and forensics.

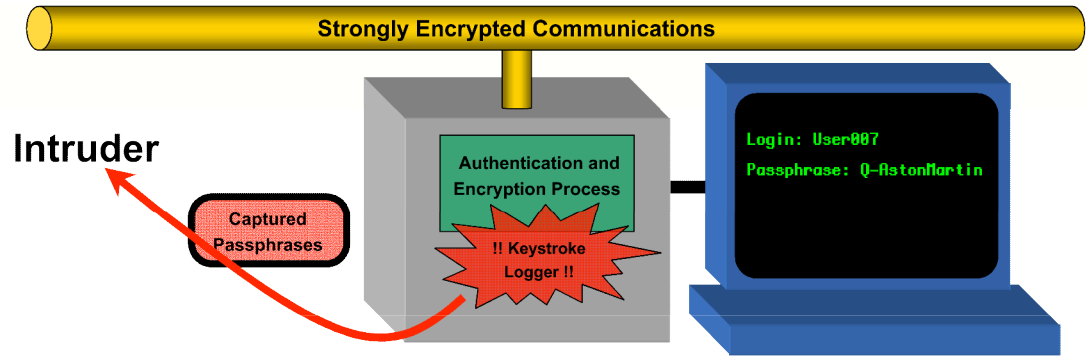

Figure 5: Keystroke loggers capture passphrases before the data stream is encrypted

Several factors adding to the complexity of the TeraGrid's security infrastructure comes from the nature of shared and delegated authority and authorization common with a federation of resources. First, a user who is granted an allocation to use TeraGrid resources must be identified and authenticated. Since it is impractical to ask new users to visit each of the nine sites in person, TeraGrid partners must trust the identification procedures performed at partner sites. Second, because the TeraGrid permits "single sign-on" capability, an authenticated user logged into one site may access the resources of another site without additional challenges. Third, since many of the largest sites maintain their own certificate authority (CA), essentially a master key for security certificates generated by a site, the policies and procedures for securing the CA are vital. Many readers may recognize that these three challenges make the TeraGrid, and nearly every other Grid system, only as secure as its weakest link; the entire TeraGrid is vulnerable to attack if one of the sites is compromised. While it is possible to design systems where sites are mutually distrustful and users must therefore obtain and maintain separate credentials for each site, many users find such systems very difficult to use. So instead, the TeraGrid addresses these risks by using additional safeguards and carefully reviewed policies.

The Encryption Is Strong, but the Server Is Weak

Today, most distributed computing infrastructures use a password or passphrase for a key step in the authentication process. To safeguard passphrases and sensitive data as it travels across the network, Grid middleware uses strong encryption. While encrypted channels have made snooping the network extremely difficult, the data is still quite vulnerable *before* it is encrypted, while it is inside the server. Furthermore, numerous tools and programs are widely available on the Internet for snatching passphrases. A passphrase can be snooped either by using a keystroke logging program, which records every key depressed on the keyboard, or by augmenting the standard authentication process, such as the SSH daemon, to copy and save all passphrases. The captured secrets are either saved on disk, for later pickup by the intruder, or mailed to a remote machine (see Figure 5). Installing the authentication daemon Trojan or keystroke logger requires privileged access to the computer; however, with the number of aging, poorly maintained office and classroom computers at universities and laboratories, vulnerable computers abound. A handful of compromised computers could yield dozens of user accounts and passphrases.

The relatively large social networks in the computational science community exacerbate the problem. Snatching a passphrase from an office computer could lead the intruder to a large supercomputer center, such as CERN, NCSA, or SDSC. From *inside* the large facility, the intruder would once again search for, and likely find, a poorly maintained or misconfigured server. At that point, the process would be repeated, with a new set of users and passphrases being captured—providing access to a new set of servers. Scientists maintain accounts on many machines, and the social network of scientists and the computers they frequent grows very quickly. This problem is not in the least specific to Grids but is common to systems with reusable passphrases and highly shared infrastructure. Like the TeraGrid, many universities, laboratories, and high-performance computing centers around the world have repeatedly suffered from these attacks.

One solution, while not impenetrable, is the use of hardware tokens during authentication. A password is entered into the pocket-watch-sized device, and a one-time access code is displayed on the unit. That code is then used for authentication to the computer server. Since the code may be used only once, or for a very small window of time, capturing the access code is not generally useful to intruders. The devices are very helpful in eliminating passphrase theft and are becoming widespread. The TeraGrid has done limited trials of the devices and is currently examining the costs and long-term support issues.

The TeraGyroid Collaboration

The TeraGrid project has collaborated with the UK e-Science Programme and the RealityGrid project on the design and implementation of national-scale production Grid systems. The UK has deployed several generations of large-scale Grid systems; the UK National Grid Service (NGS) is similar in design to the TeraGrid. In September 2003, the UK e-Science Programme initiated and funded an effort to link the UK and U.S. Grid systems for demonstration at the Supercomputing 2003 conference. The project was designed to explore trans-Atlantic optical networking, Grid software compatibility, security, and the performance of linked Grid systems. The RealityGrid system, which includes simulation, visualization, checkpoint restart, and computational steering capabilities, was chosen as the application framework for experiment. The simulation, TeraGyroid, was a lattice-Boltzmann code exploring defect dynamics in amphiphilic liquid crystals.

Linking the two Grid systems required solving four significant problems. The first problem was fast networking. British Telecom helped construct the UKLight optical path to Chicago. There, the link was patched into the TeraGrid Chicago hub, connecting the UK resources at Manchester and Daresbury/Edinburgh to the TeraGrid. Second, a joint security plan was developed. The UK scientists were already using identities digitally signed by a UK certificate authority. To grant them access to the

U.S. resources, the TeraGrid reviewed the UK e-Science security plans and then accepted the UK certificate signing authority. Third, the RealityGrid code had to be ported to the TeraGrid supercomputer platforms and Grid middleware. Finally, the visualization and computational steering components of the RealityGrid framework needed to be joined with the Access Grid, a collaborative environment supporting group-to-group interactions across the Grid. The Access Grid was used by TeraGyroid to view and discuss ongoing computations with scientists on different continents. The project was able to run the largest simulation of its kind using the joint resources and won the HPC Challenge Award for "Most Innovative Data-Intensive Application at SuperComputing 2003." By the middle of 2004, the NGS and TeraGrid agreed to keep the systems linked and maintain a permanent network connection from the UK to the TeraGrid and continue to accept the UK certificate authority. Furthermore, research teams that span the United States and the UK have received resource allocations on the TeraGrid, fostering joint work. The next steps will be to better synchronize and standardize the middleware used by the two systems. Currently, the software differences between the Grids are not easily masked.

Conclusions

The TeraGrid project created a unified nationwide Grid for high-performance computing. The infrastructure enables the computational science and engineering communities to combine and share resources across large, diverse developments teams and simplified access with improved service. Using the TeraGrid, teams are studying everything from the origins of the cosmos to nanotechnology. While the TeraGrid was being constructed, several important contributions were made to the field of distributed, computational infrastructure:

- *A transcontinental high-speak optical network:* The TeraGrid network is among the most powerful of its kind in the world. Combined with enhanced tools for parallel data transfer, the TeraGrid has provided ultra-fast data movement that makes it easier to run remote computation against local data.
- *Distributed accounting, account management, and operations:* The TeraGrid built a distributed accounting system that allows for remote account creation and a single accounting currency for compute jobs. The system permits sites to maintain their existing job accounting infrastructure and converts local data to a shared format for the central database. A jointly run distributed operations center shares a trouble ticket system and the responsibility to resolve problems.
- *A unified security plan:* TeraGrid has built a distributed security team and policy framework that plans for security and responds to incidents. The team maintains security incident response protocols and playbooks, policies for accepting and revoking certificates, media communication plans, and tools for users to manipulate their security credentials.
- *Coordinated Grid middleware:* The TeraGrid has carefully specified the software and services environment provided to users. CTSS is based on standards where they are appropriate; for example, CTSS includes MPI, SSH, and Fortran 90 for compute platforms. CTSS also presents a uniform set of environment settings that permit users to find and use TeraGrid software resources easily. The TeraGrid is also unique to most Grid systems in not requiring identical binary installations of software—instead the TeraGrid is more open and uses standard protocols or compatible versions where possible.
- *A rigorous testing and verification framework:* The TeraGrid invested heavily in continuous testing of the TeraGrid environment. Inca testing and verification includes tests not only for specific components in the CTSS environment but also for correct behavior and overall performance. Grid middleware remains extremely fragile and difficult to properly configure. However, rigorous testing regimes can help improve the stability of Grid systems even in the presence of rapidly shifting and evolving standards and components. Building a production-quality distributed computing infrastructure is impossible without careful and aggressive testing and verification.

As the TeraGrid moves forward, several new directions and enhancements are planned for the infrastructure. One area of recent interest is "on-demand" computation for time-critical decision support. In several fields, high-performance computing has produced applications that can be used for time-critical

decision support. For example, the LEAD project uses radar data and weather models to predict severe weather. In order to be useful for public safety, the computation must complete before the severe weather strikes. An on-demand framework for high-priority jobs must be developed. Another example comes from Los Alamos National Laboratory. EpiSims is a epidemiological simulation for modeling the spread of epidemics at the level of individuals in a large urban region with realistic contact patterns and disease transmission characteristics. TeraGrid is a perfect host for these types of computational challenges, provided a scheduling and resource framework that can be augmented to support on-demand computation.

Moving the current model, which supports individuals and small teams, to supporting broad science communities is another area for development within the TeraGrid. Currently, nearly all the resource allocations go to Principal Investigators who apply on behalf of a small team of application scientists. By engaging an entire community, the TeraGrid can provide tools and compute cycles to large groups that share applications and standardized tool chains. For example, the GridBlast portal supports a tool chain that can perform functional and comparative genomics. Developing the policies, allocation procedures, and infrastructure to support these communities would let TeraGrid essentially become a wholesaler, providing resources to a science community that would organize its users and applications.

In addition to these areas of growth, the TeraGrid must strengthen collaboration with UK, Japanese, and European Grid systems that are building production high-performance infrastructures. The RealityGrid collaboration was an important first step, but additional work is needed to make the software environments more compatible, share and exchange resource allocations, and improve security.

References

Allan, R., Chohan, D., Wang, X. D., Richards, A., McKeown, M., Colgrave, J., Dovey, M., Baker, M. & Fisher, S. 2003. Building the e-Science Grid in the UK: Middleware, applications and tools deployed at level 2. In *Proc. all hands meeting*, pp. 556–562, Nottingham, UK

Baru, C., Moore, R., Rajasekar, A. & Wan, M. 1998. The SDSC storage resource broker. Presentation at *8th annual IBM Centers for Advanced Studies conference*, Toronto, Canada.

Brunner, R. J., Djorgovski, S. G., Prince, T. A. & Szalay, A. S.: 2002. The National Virtual Observatory. In *Proc. extragalactic gas at low redshift* (eds: J. S. Mulchaey & J. T. Stocke), p. 383, San Francisco: Astronomical Society of the Pacific.Conference.

Foster, I. & Kesselman, C. 1998. Globus: a metacomputing infrastructure toolkit. *Int. J. Supercomputer App*s. **11** (2) 115–129.

Frey, J., Tannenbaum, T., Foster, I., Livny, M. & Tuecke, S. 2001. Condor-G: A computation management agent for multi-institutional Grids. *Cluster Computing* ***5*** (3) 237–246.

Novotny, J., Tuecke, S. & Welch, V. 2001. An online credential repository for the Grid: MyProxy. In *10th IEEE int. symp. on high performance distributed computing*, IEEE Computer Society Press.

Sullivan III, W. T., Werthimer, D., Bowyer, S., Cobb, J., Gedye, D. & Anderson, D. 2001. A new major SETI project based on Project Serendip data and 100,000 personal computers. In *Proc. 5th international conference on bioastronomy* (eds. C. B. Cosmovici, J. Bowyer & D.Werthimer) Bologna, Italy: Editrice Composition.

Xue, M. D.-H., Wang, J.-D., Gao, K. & Droegemeier, K. K. 2003. The Advanced Regional Prediction System (ARPS), storm-scale numerical weather prediction and data assimilation. *Meteor. Atmos. Physics*, **82**, 139–170.