CTWatch QUARTERLY

ISSN 1555-9874

Volume 2 Number 2 **May 2006**

# Designing and Supporting Science-Driven Infrastructure

GUEST EDITORS: **FRAN BERMAN** AND **THOM DUNNING**



AVAILABLE ON-LINE AT
**http://www.ctwatch.org/quarterly/**

Cyberinfrastructure Technology Watch
http://www.ctwatch.org/

INTRODUCTION

# Designing and Supporting Science and Engineering-Driven Infrastructure

Fran Berman
Director, San Diego Supercomputer Center

Thom H. Dunning, Jr.
Director, National Center for Supercomputing Applications
Professor and Distinguished Chair for Research Excellence, Department of Chemistry, University of Illinois at Urbana-Champaign

The seminal 2003 Report from the *Blue Ribbon Advisory Panel on Cybeirnfrastructure* (the "Atkins Report") states that

> "The term **infrastructure** has been used since the 1920's to refer collectively to the roads, power grids, telephone systems, bridges, rail lines, and similar public works that are required for an industrial economy to function. Although good infrastructure is often taken for granted and noticed only when it stops functioning, it is among the most complex and expensive thing that society creates. The newer term **cyberinfrastructure** refers to infrastructure based upon distributed computer, information and communication technology. If **infrastructure** is required for an **industrial** economy, then we could say that **cyberinfrastructure** is required for a **knowledge** economy."

No-one would question that bridges, roads, and telephones require maintenance, regular upgrades, support staff, long-range planning, continuous monitoring, and other items, just as no-one would question their importance to the successful functioning of society.

In the cyber-world, compute, data, networking, software, and other kinds of infrastructure are equally important as enablers for education and new discovery. Information technologies are ubiquitous to modern scientists and engineers, and the ability to utilize relevant resources in a coordinated way is critical for progress.

Analogous to public infrastructure, cyberinfrastructure requires maintenance, regular refresh/upgrade, support staff, long-range planning, monitoring, and other items. For our nationally allocated resources, help-desks, bug fixes, community codes, public data, and other components are all part of the researcher's toolkit. Unlike research that is characterized by innovation, flexibility, and risk, successful cyberinfrastructure must be characterized by usefulness, usability, functionality, and stability.

In this issue of *Cyberinfrastructure Technology Watch*, we look behind the scenes to understand what is required to develop and deploy successful cyberinfrastructure for compute, data, software, and grids. We have asked a distinguished set of colleagues and authors – some of the most experienced individuals in the community – to contribute to this issue on the "real" costs of cyberinfrastructure. We thank our colleagues for their efforts, and hope that *CTWatch* readers enjoy this glimpse behind the scenes of cyberinfrastructure.

# Creating and Operating National-Scale Cyberinfrastructure Services

Charlie Catlett
Pete Beckman
Dane Skow
Ian Foster
The Computation Institute, University of Chicago and Argonne National Laboratory

## 1. Introduction

The term "cyberinfrastructure" is broadly defined to include computer applications, services, data, networks, and many other components supporting science.[1] Here we discuss the underlying resources and integrative systems and software that together comprise a grid "facility" offering a variety of services to users and applications. These services can range from application execution services to data management and analysis services, presented in such a way that end-user applications can access these services separately or in combination (e.g., in a workflow).

We use the TeraGrid[2] project to illustrate the functions and costs of providing national cyberinfrastructure. Developed and deployed in its initial configuration between 2001 and 2004, the TeraGrid is a persistent, reliable, production national facility that today integrates eighteen distinct resources at eight "resource provider" facilities.[3] This facility supports over 1000 projects and several thousand users (Fig. 1) across the sciences. TeraGrid architecture, planning, coordination, operation, and common software and services are provided through the Grid Infrastructure Group (GIG), led by the University of Chicago. TeraGrid staff work with end-users, both directly and through surveys and interviews, to drive the technical design and evolution of the TeraGrid facility in support of science. In addition, TeraGrid is developing partnerships with major science facilities and communities to provide needed computational, information management, data analysis, and other services and resources, thus allowing those communities to focus on their science rather than on the creation and operation of services.

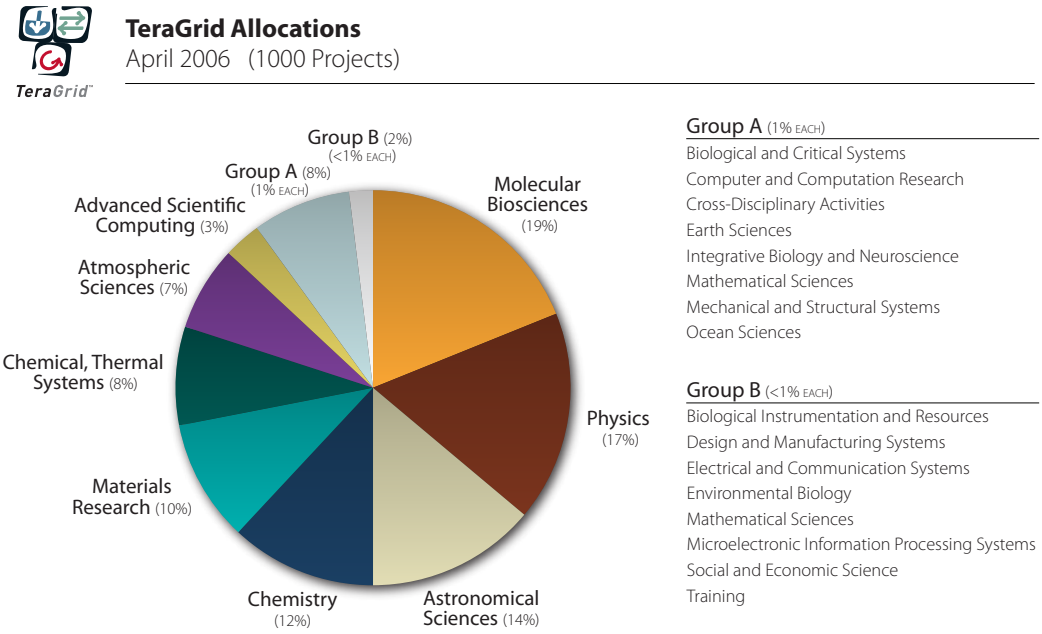


Figure 1. TeraGrid allocations by science discipline, April 2006 (1000 projects). Data from David Hart, SDSC.

[1] Atkins, D.E., Droegemeier, K.K., Feldman, S.I., Garcia-Molina, H., Klein, M.L., Messina, P., Messerschmitt, D.G., Ostriker, J.P. and Wright, M.H. "Revolutionizing Science and Engineering Through Cyberinfrastructure: Report of the National Science Foundation Blue-Ribbon Panel on Cyberinfrastructure," 2003.

[2] The TeraGrid 2006 - http://www.teragrid.org/

[3] TeraGrid Resource Providers are Argonne National Laboratory / University of Chicago, Indiana University, the National Center for Supercomputing Applications, Oak Ridge National Laboratory, the Pittsburgh Supercomputing Center, Purdue University, the San Diego Supercomputer Center, and the Texas Advanced Computing Center.

TeraGrid supports a variety of use scenarios, ranging from traditional supercomputing to advanced Grid workflow and distributed applications. In general terms, TeraGrid emphasizes two complementary types of use. TeraGrid "Deep" involves harnessing TeraGrid's integrated high-capability resources to enable scientific discovery that would not otherwise be possible. TeraGrid "Wide" is an initiative that is adapting TeraGrid services and capabilities to be readily used by the broader scientific community through interfaces such as web portals and desktop applications. All of these use scenarios—even traditional supercomputing users—benefit from the common services that are operated across the participating organizations, such as uniform access to storage, common data movement mechanisms, facility-wide authentication, and distributed accounting and allocations systems that provide the basis for authorization.

Creating and operating a grid facility involves integrating resources, software, and user support services into a coherent set of services for users and applications. Resources are explored by Roskies,[4] while Killeen and Simon[5] discuss user and community support. We discuss here the software infrastructure and policies required to integrate these diverse components to create a persistent, reliable national-scale facility. While the federation of multiple, independent computing centers requires carefully designed federation, governance, and sociological policies and processes, in this article we focus only on the functional and technical costs of operating a national grid infrastructure.

[4] Roskies, R., Zacharia, T. "Designing and Supporting High-end Computational Facilities," CTWatch Quarterly 2(2): May 2006.

[5] Killeen, T. L., Simon, H. D. "Supporting National User Communities at NERSC and NCAR," CTWatch Quarterly 2(2): May 2006.

## 2. Software Infrastructure

Software components in a grid facility include science applications, grid middleware, infrastructure support services, and mechanisms to integrate community-developed systems we call "Science Gateways." If we define the fundamental components of infrastructure to be those that have the longest useful lifespan, then software is clearly the critical investment. While particular platforms (e.g., x86) may have long lifetimes, individual high-end computational resources have a useful lifespan of perhaps five years. In contrast, many components of our software infrastructure are already 10 years old. For example, TeraGrid deployed the Globus Toolkit[6] nearly five years ago (it was not new at the time), and our expectation is that this software will be integral for the foreseeable future. Similarly, scientific communities have invested several years in building software infrastructure – tools, databases, and web portals for example – for their communities. Science application software and the tools for developing, debugging and managing that software are often even older. As we consider costs and investments for integrating grid facilities, it is essential that we leverage these investments.

[6] Foster, I. "Globus Toolkit Version 4: Software for Service-Oriented Systems," IFIP International Conference on Network and Parallel Computing, 2005, Springer-Verlag LNCS 3779, 2-13.

### 2.1 Middleware Software and Services

The vast majority of scientific grid facilities rely heavily on a common core set of middleware systems, such as the Globus middleware (which includes numerous components, such as GridFTP for data transfer, GRAM for job submission, Grid Security Infrastructure, and the credential management software MyProxy[7]) and a variety of related tools such as the Condor scheduling system[8] and the verification and validation suite Inca.[9] The development and wide-scale adoption of these components has been made possible by substantial investments by DOE, NSF, and other agencies in the U.S. and abroad. In particular, NSF's investment of roughly $50M in the NSF Middleware Initiative (NMI) program[10] over the past five years has played a key role in developing and "hardening" these and other software systems such that they can be reliably used in grid facilities, as evidenced by their widespread adoption world-wide in hundreds of grid projects and facilities. For example, the NMI GRIDS Center[11] has supported the development,

[7] Novotny, J., Tuecke, S. and Welch, V. "An Online Credential Repository for the Grid: MyProxy," 10th IEEE International Symposium on High Performance Distributed Computing, San Francisco, 2001, IEEE Computer Society Press.

[8] Litzkow, M. and Livny, M. "Experience with the Condor Distributed Batch System," IEEE Workshop on Experimental Distributed Systems, 1990.

[9] Smallen, S., Olschanowsky, C., Ericson, K., Beckman, P. and Schopf, J.M. "The Inca Test Harness and Reporting Framework," SC'2004 High Performance Computing, Networking, and Storage Conference, 2004.

[10] NSF Middleware Initiative (NMI), 2006 - http://www.nsf-middleware.org/

[11] NSF Middleware Initiative (NMI) Grid Research Integration Development and Support (GRIDS) Center, 2006 - http://www.grids-center.org/

integration testing, and packaging of many components. This work has reduced the complexity of creating a basic grid system and greatly simplified updating systems that adopted earlier versions of software. Additional investments of tens of millions of dollars has been made worldwide in grid deployment projects that have contributed to the maturation of these software systems, the development of tools for particular functions, and the pioneering of the new application approaches enabled by TeraGrid-class facilities. For example, the TeraGrid project invested roughly $1M in the initial design and development of the Inca system, which is one of many such components that are available today through the NMI program.

Continued investment in middleware capabilities development, through programs like NMI, is critical if we are to deliver on the promise of cyberinfrastructure. Major grid facilities like TeraGrid, and the user-driven application and user environment projects that build on those facilities, typically involve a two-year development schedule and a five-year capability roadmap, both of which rely on the progression of capabilities from research prototypes to demonstration systems to supportable software infrastructure.

### 2.2 Science Gateways

In parallel with NMI over the past several years, other programs within NSF, DOE, NIH, and other agencies have provided funding to bring together software engineers and computational scientists to create software infrastructure aimed at harnessing cyberinfrastructure for specific disciplines. For example, the Linked Environments for Atmospheric Discovery[12] project is creating an integrated set of software and services designed for atmospheric scientists and educators. Similar cyberinfrastructure has been created in other disciplines such as high energy and nuclear physics,[13] [14] [15] fusion science,[16] earth sciences,[17] [18] astronomy,[19] [20] nanotechnology,[21] bioinformatics,[22] and cancer research and clinical practice.[23]

In the TeraGrid project we have formed a set of partnerships around the concept of "Science Gateways," with the objective of providing TeraGrid services (e.g., computational, information management, visualization, etc.) to user communities through the tools and environments they are already using, in contrast to traditional approaches that require the user to learn how to use the Grid facilities directly. The most common presentation of these community-developed cyberinfrastructure environments is in the form of web portals, though some provide desktop applications or community-specific grid systems instead of or in addition to.

We have partnered in the TeraGrid project not only with gateway providers but also with other grid facilities to identify and standardize a set of services and interaction methods that will enable web portals and applications to invoke computation, information management, visualization, and other services. While still in the early stages, the TeraGrid Science Gateways program has catalyzed a new paradigm for delivering cyberinfrastructure to the science and education community, with a scalable wholesale/retail relationship between grid facilities and gateway providers. Additional benefits to this model include improved security architecture (offering targeted, restricted access to users rather than open login access) and collaboration support (community members can readily share workflows, tools, or data through and among gateway systems).

[12] Droegemeier, K. et al, "Linked Environments for Atmospheric Discovery (LEAD): Architecture, Technology Roadmap, and Deployment Strategy," 21st Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology, 2005, American Meteorological Society.

[13] Open Science Grid (OSG), 2006 - http://www.opensciencegrid.org/

[14] Avery, P. and Foster, I. "The GriPhyN Project: Towards Petascale Virtual Data Grids," 2001 - http://www.griphyn.org/

[15] Avery, P., Foster, I., Gardner, R., Newman, H. and Szalay, A. "An International Virtual-Data Grid Laboratory for Data Intensive Science," Technical Report GriPhyN-2001-2, 2001 - www.griphyn.org

[16] Schissel, D.P., Keahey, K., Araki, T., Burruss, J.R., Feibush, E., Flanagan, S.M., Foster, I., Fredian, T.W., Greenwald, M.J., Klasky, S.A., Leggett, T., Li, K., McCune, D.C., Lane, P., Papka, M.E., Peng, Q., Randerson, L., Sanderson, A., Stillerman, J., Thompson, M.R. and Wallace, G. "The National Fusion Collaboratory Project: Applying Grid Technology for Magnetic Fusion Research," Workshop on Case Studies on Grid Applications, 2004.

[17] GEON: The Geosciences Network, 2006 - http://www.geongrid.org/

[18] Bernholdt, D., Bharathi, S., Brown, D., Chanchio, K., Chen, M., Chervenak, A., Cinquini, L., Drach, B., Foster, I., Fox, P., Garcia, J., Kesselman, C., Markel, R., Middleton, D., Nefedova, V., Pouchard, L., Shoshani, A., Sim, A., Strand, G. and Williams, D. "The Earth System Grid: Supporting the Next Generation of Climate Modeling Research," Proceedings of the IEEE, 93 (3). 485-495. 2005.

[19] National Virtual Observatory, 2006 - http://www.us-vo.org/

[20] Szalay, A. and Gray, J. "The World-Wide Telescope," Science, 293. 2037-2040. 2001.

[21] Nanotechnology Simulation Hub (NanoHub), 2006 - http://www.nanohub.org/

[22] Ellisman, M. and Peltier, S. "Medical Data Federation: The Biomedical Informatics Research Network," The Grid: Blueprint for a New Computing Infrastructure (2nd Edition), Morgan Kaufmann, 2004.

[23] Cancer Bioinformatics Grid (caBIG), 2006 - http://cabig.nci.nih.gov/

## 3. Integration: Building a National-Scale Grid Facility

The creation of a grid facility involves the integration of a set of resource providers. A coherent grid facility must leverage software infrastructure, as described above, to provide a set of common services, a framework that allows for exploitation of unique facilities, and the infrastructure needed to coordinate the efforts of the resource providers in support of users. Common services include operations centers, network connectivity, software architecture and support, planning, and verification and validation systems. Facility-wide infrastructure includes components such as web servers, collaboration systems, the framework for resource management policy and processes, operation coordination, training documentation and services, and software repositories. Underlying Grid middleware software provides common services and interfaces for such functions as authentication and authorization, job submission and execution, data movement, monitoring, discovery, resource brokering, and workflow.

For scientific computing, and in particular high-performance computing, the fact that a user can reliably expect the Unix operating system as the standard environment on almost all major shared resources has been a boon to scientists making persistent software investments. Internet connectivity and basic services such as SSH and FTP have similarly become standard offerings. A grid facility aims to provide services that allow for resources to be aggregated, such that applications hosted on various resources can be combined into a complex workflow. Such a set of services, operated within a single organization, would be merely complex. Providing these services across a collection of organizations adds policy, social, coordination, and other integration requirements that exceed the complexity of the grid middleware itself.

Addressing these requirements to create and manage a national-scale grid environment requires the creation and operation of both organizational and technical integration services. We do not attempt to prescribe organizational structures within which these functions reside. However, we do make several observations. First, a grid facility requires close collaboration and cooperation among all participating organizations, each of which provides one or more functions and services as part of the overall facility. Second, despite the fact that each participating resource provider shares the goal of creating and operating a high-quality grid facility, it is necessary to identify specific responsibilities for coordinating and providing common services. In most grid projects, this function is performed by a system integration team that coordinates and plans common services, providing these services directly and through partner organizations.

In the rest of this section, we use the TeraGrid to illustrate the specific functions and costs required to provide national cyberinfrastructure. For each functional section, we discuss the scope of work as well as the approximate staffing levels, both in the system integration team (the GIG) and at the resource provider facilities. We use units of "full time equivalents" or "FTE" to measure effort because most staff members are employed partially on TeraGrid funding and partially on other institutional funding.

### 3.1 Software and Resource Integration and Support

The TeraGrid software environment involves four areas. **Grid middleware**, including the Globus Toolkit, Condor, and other tools, provides capabilities for harnessing TeraGrid resources as an integrated system. These Grid middleware components are deployed, along with libraries, tools, and virtualization constructs, as the **Coordinated TeraGrid Software and Services (CTSS)** system, which provides users with a common development and runtime environment across heterogeneous platforms. This common environment lowers barriers users encounter in exploiting the diverse capabilities of the distributed TeraGrid facility to build and run applications. A software deployment **validation and**

**verification** system, Inca, continuously monitors this complex environment, providing users, administrators, and operators with real-time and historical information about system functionality. In addition, users are provided with login credentials and an allocations infrastructure that allows a single allocation to be used on any TeraGrid system through the Account Management Information Exchange (AMIE)[24] **account management and distributed accounting** system.

[24] Account Management Information Exchange (AMIE), 2006 - http://scv.bu.edu/AMIE/

These four components must work seamlessly together, combined with related administrative policies and procedures, to deliver TeraGrid services to users. Software integration efforts must ensure that these components can be readily deployed, updated, and managed by resource provider staff, while working with science partners to both harden and enhance the capabilities of the overall system and with the NMI project to implement an independent external test process.

Increasingly, TeraGrid is also providing service-hosting capabilities that allow science communities to leverage the operational infrastructure of this national-scale grid. For example, data and collections-hosting services are provided as part of the TeraGrid resource provider activities at SDSC. Users may request storage space, specifying their desired access protocols, ranging from remote file I/O via a wide area parallel filesystem to GridFTP[25] and Storage Resource Broker (SRB).[26] Similarly, communities are provided with software areas on all TeraGrid computational resources, thus enabling community-supported software stacks and applications to be deployed TeraGrid-wide.

[25] Allcock, B., Bresnahan, J., Kettimuthu, R., Link, M., Dumitrescu, C., Raicu, I. and Foster, I., "The Globus Striped GridFTP Framework and Server". SC'2005, 2005.

[26] Baru, C., Moore, R., Rajasekar, A. and Wan, M. "The SDSC Storage Resource Broker," 8th Annual IBM Centers for Advanced Studies Conference, Toronto, Canada, 1998.

A general-purpose facility such as TeraGrid must evolve constantly in concert with the changing and growing needs and ideas of its user community. Sometimes the need for a new capability or the improvement of an existing one will be obvious from operational experience or groundswell requests from the user community. In other cases, multiple competing ideas may arise within particular communities/subsets of the facility that must either be replaced by a new common component, or integrated into a coherent system. TeraGrid services are defined as part of the CTSS package, with major releases at roughly six-month intervals used to introduce new capabilities.

The costs of integrating, deploying, and operating these software systems can be significant. The TeraGrid project applies 10 FTEs to the tasks of integrating, verifying, validating, and deploying new capabilities. This staff works with 42 **resource integration and support** FTEs from the resource provider facilities. The latter staff is responsible for the support and administration of the specific computational, information management, visualization, and data resources operated by resource provider facilities.

### 3.2 Coordinating User Support and User Support Infrastructure

User support is best done in a manner that can fully exploit all available human connections to users and their problem domains. The most frequent model is to have the user support staff local to the resource providers. This model is motivated in part by the historical organization of computing centers as vertically integrated, standalone facilities, and in part by the fact that close connection to the users and their issues is important to the centers, providing vital information for tuning, improving and designing next generation facilities.

TeraGrid leverages this model, coordinating the support staff across the sites to provide a set of support programs that give users a "one stop shop" whose major function (beyond basic "first

aid") is to establish the connection between the user and the appropriate local support. This approach also allows us to draw on the expertise and availability of peers across the full organization.

This coordinated, leveraged approach is essential when supporting a user community in the context of a distributed grid facility, where services and applications involve multiple components. Diagnosing and tuning applications in such an environment often requires the engagement of experts from multiple organizations. At the same time, it is important that a single responsible party "own" getting a solution to the user. Often, providing a modest amount of focused attention, while drawing on specialists across the facility, allows researchers to make rapid substantial progress in the efficiency and capabilities of their applications.

TeraGrid user support services comprise three FTEs who provide central coordination and 25 applications support and consulting FTEs from the resource provider facilities. A particular benefit to this distributed teaming approach is that TeraGrid can draw on a much more diverse group of experts than can be found in any single facility.

The TeraGrid GIG is also creating a team of experts whose role currently is to integrate a set of prototype science gateways. Consisting of 10 FTE located at eight science gateway sites, this distributed support team will shift within 12-18 months from primarily integrating prototypes to becoming a general support team for the dozens of science gateways that we anticipate will emerge from these early pioneering efforts. Complementing the direct end-user support team, this team's customers will be user support and technical staff associated with science gateways.

### 3.3 External Communications, Training and Documentation

As with a single-site facility, a national cyberinfrastructure requires focused effort on communications to key groups, including end-users, funding agencies, and other stakeholders. Each resource provider within a national grid facility will provide documentation and training for the resources and services locally provided, and these materials must be proactively integrated, in a similar fashion to the services and resources themselves. This tasks requires an overall communication architecture that provides structure and common interfaces and formats for the training and documentation materials, along with the curation – the analog to software verification and validation – of the overall systems.

TeraGrid coordinates these areas with two FTEs who work with three FTEs at resource provider facilities as well as the external relations, education, and training staff at those facilities (but not dedicated to TeraGrid).

A key strategy for not only communication but also user support and simplifying the use of TeraGrid is a user portal program that provides users with a web-based, customizable interface for training, documentation, and common user functions such as resource directories, job submission and monitoring, and management of authorization credentials across TeraGrid. The user portal project involves two FTEs who work closely with the communications, training, and documentation staff.

### 3.4 Operational Services

While largely transparent to end-users, any national grid facility must be supported by a deep foundation of operational infrastructure. This need is particularly important for facilities such as TeraGrid that operate national-scale resources, purchased and supported on behalf of government agencies, where accountability for the use of those resources is required, coupled with an open peer-

review process for allocating access to the resources. Operational services discussed here also include networking, security coordination, and an operations center.

*Resource Allocation and Management*

Many national-scale grid consortia operate "best-effort" services that provide access to excess capacity to stakeholder user groups. In contrast, TeraGrid operates resources on behalf of broad national communities, and these resources are allocated by formal processes. Specifically, resources are allocated by a peer-review committee that meets quarterly to review user requests for allocations. (Allocations are specified in service units, analogous to CPU hours.) The mechanisms needed to support this nationally peer-reviewed system include a distributed accounting system that works in concert with authentication and authorization systems to debit project allocations according to use by users authorized by the principal investigator of the given project. In addition, support for the allocation review process itself requires a proposal request and review infrastructure, databases for users and usage, and information exchange systems for usage data and user credentials. The TeraGrid has obtained much of this infrastructure from its predecessor, the NSF Partnerships for Advanced Computational Infrastructure (PACI) program, in which several million dollars of software development was invested during the past decade.

The operation of the TeraGrid resource allocation and management infrastructure requires four GIG FTEs for coordination along with seven FTEs at resource provider facilities to support the various databases and proposal support systems, and to perform local accounting integration with the distributed TeraGrid system.

*Security Coordination*

Security management in a national grid facility requires a high degree of coordination among security professionals at many sites. TeraGrid security coordination is based on a set of agreed-upon policies ranging from minimum security practices to change management and protocols for incident response and notification.

The GIG team provides coordination of the distributed security team for general communication, incident response management, and analysis of the security impact of system changes (e.g., software, new systems, etc.). However, the provision of distributed authentication and authorization services for individual users and groups (or "virtual organizations"[27]), as is required in grid facilities, is also a significant part of the security coordination effort.

[27] Foster, I., Kesselman, C. and Tuecke, S. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," International Journal of Supercomputer Applications, 15 (3). 200-222. 2001.

Security coordination across TeraGrid requires two GIG FTEs working with three FTEs at resource provider sites, with participation from additional security operations staff from each resource provider organization. While participation in a national grid security coordination team requires investment of time on the part of local security staff, the benefits to the site are high in terms of training, assistance, and early notification of events that might impact the local site.

*Networking*

Many national-scale grid facilities rely on existing Internet connectivity. In contrast, TeraGrid operates a dedicated network. Irrespective of the networking strategy, effort is needed to optimize services over networks between resource provider locations, particularly with respect to data movement over high bandwidth-delay product networks. In addition, distributed applications and services often require assistance from networking experts at multiple sites. Thus, a national-scale grid facility such as TeraGrid requires a networking team consisting of contacts

from each resource provider site. As with the security team, the benefits to the site far outweigh the time-investment on the part of local networking staff.

In the case of TeraGrid, this component of the support infrastructure comprises a network architect/coordinator within the GIG to oversee the networking team, which includes five FTEs from resource provider facilities along with general networking contacts at all sites. The networking working group coordinates the operation of the TeraGrid network. Participants also assist in user support, such as diagnosing problems and optimizing performance of distributed services and applications.

*Operations*

TeraGrid provides a distributed operations center, leveraging the 24/7 operations centers at two of the resource provider facilities (NCSA and SDSC) to provide around-the-clock support. The distributed 24/7 operations center plays several essential roles in the TeraGrid facility, including the management of a common trouble-ticket system and ongoing measurement of key metrics related to the health and performance of the facility. TeraGrid operations requirements also include management of the distributed accounting system, which involves the collection of usage information into a central usage database. The TeraGrid GIG funds two FTEs for various aspects of operations and two FTEs at resource provider facilities.

### 3.5 Catalytic Processes: Policy, Planning and Internal Coordination

The creation of a national-scale team comprised of individuals from multiple independent institutions requires careful attention to collaboration systems and processes in support of virtual, and distributed, teams. Two FTEs within the TeraGrid GIG maintain infrastructure (e.g., CVS repository, discussion forums, wiki and bugzilla servers) that is used both for day-to-day collaboration and to "curate" important project data. For coordination of activities, TeraGrid relies on two types of virtual teams. *Working groups* are persistent groups of TeraGrid staff with a common mission, such as supporting TeraGrid software, networks, or accounting systems. These working groups are complemented with short-term planning teams called *requirements analysis teams* (affectionately, "RATs"). Working groups typically involve key members from each resource provider site and coordinators from the GIG and meet regularly on an ongoing basis. RATs are generally smaller (4-6 members) and work on a particular issue for 6-8 weeks to produce a recommendation or proposal for new policy or projects.

The resources and integrative software and services that make up a national-scale grid facility define one axis of its operations. However there is also a distinct institutional axis, which is where decisions are made regarding the facility's operations, policies, and major changes to its services, resources, and software. TeraGrid formalizes these latter processes in terms of a numbered, citable, persistent document series, not unlike those used by standards bodies. The initial document[28] in the series lays out the roles and responsibilities of TeraGrid's GIG and resource provider partners as well as a decision-making process.

[28] Catlett, C., Goasguen, S. and Cobb, J. "TeraGrid Policy Management Framework," TeraGrid Report TG-1, 2006.

While top-down, hierarchical management is feasible in a single organization, a federation of interdependent peer organizations requires a different model. At the same time, while democratic processes may work for loose collaborations, they are not appropriate for operation of a production facility. TeraGrid decision-making relies on consensus among representatives of each resource provider, under the leadership of the principal investigator of the GIG who serves as overall TeraGrid project director. This team of resource provider and GIG principals, called

the Resource Provider Forum, meets weekly in an open Access Grid session and quarterly for face-to-face review and planning.

## 4. Summary and Conclusions

Figure 2 shows how the TeraGrid cyberinfrastructure facility allocates staff to provide high-capability, high capacity, high-reliability computational, information management, and data analysis services on a national scale. Approximately 25% of the staff are allocated to common integration functions (TeraGrid GIG) and 75% to resource provider facility functions. User support and external communications are emphasized at similar levels in both the resource provider efforts and the common GIG effort. GIG effort is the bulk of the software, policy and management, and operational services; resource provider effort is the bulk of the resource integration and support and functions. Note that even the "central" functions are distributed: the common services are largely staffed in a distributed fashion at the resource provider sites. TeraGrid's GIG, operated by the University of Chicago, relies on subcontracts with resource provider facilities for more than 2/3 of the GIG staff, making even the common services team a distributed enterprise. What is important is that this GIG staff, and the services that it provides, is coordinated by a single entity.

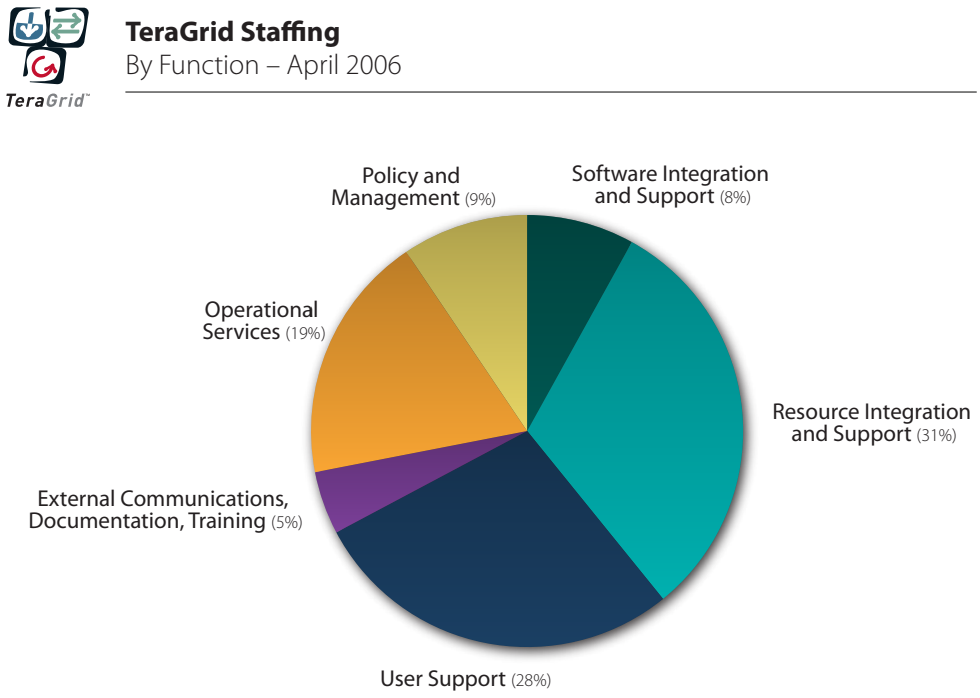


Figure 2. TeraGrid staffing distribution by functional area, April 2006.

Although these numbers will differ in the particular areas from one national grid project to another, we believe that they are representative of the general balance of requirements, both among different functions and between "common" or centrally-provided services and those provided by resource provider facilities.

# Designing and Supporting High-end Computational Facilities

Ralph Roskies
Pittsburgh Supercomputing Center

Thomas Zacharia
Oak Ridge National Laboratory

In this article, we outline the types of activities required (and an estimate of their cost) in designing and supporting high-end computational facilities. The major categories are facility costs, system software, and the human effort involved in designing and keeping the systems running. This discussion does not include any costs associated with direct user support, application software, application support, or for the development of new technology. Nor does it include the networking issues related to connecting outside the machine room. Those are covered elsewhere in this volume.

## Facility Issues

The principal points to be included in planning an HPC facility are sufficient space, power, and cooling. Equally important, but often more easily amenable to improvement, are physical security, water and fire protection, pathways to the space, and automatic monitoring systems.

In the provision of space there is more to consider than the required number of square feet. This is especially true for today's air-cooled clusters, which were not designed to be used together in the large quantities found in leading HPC centers. Today's dense, air-cooled systems require large volumes of air for cooling. The size of the plenum under the floor, i.e. the area between the solid subfloor and the bottom of the raised floor tile is an important measure of the ability to deliver adequate air. Distribution is also an issue. Masses of under-floor cable tend to cause air dams which impede the ability to deliver air where it is needed. Conversely, moving large volumes of air through a barely adequate plenum will tend to cause streamlining, particularly when vents are located close to air handling units. Optimal location of the air handling units within the space often seems counter-intuitive. For example, one might think that placing air handlers close to the machine is better and more efficient. But that is likely to cause problems with streamlining and result in low pressure areas. Establishing the correct flow of air is an iterative process no matter what your CFD study says. These issues get a lot simpler with liquid-cooled systems.

There are also many mundane problems to attend to. Subfloors should be sealed to prevent cement dust from proliferating. Floor drains are needed for disposal of condensing moisture from air handler coils. Floor tiles should be carefully selected to avoid the problem of "zinc whiskers" the dispersion of tiny metallic slivers from the undersides of older tiles that cause seemingly random hardware reliability problems. Since computer equipment, air handlers, PDUS, etc. are both large and heavy, it is of great benefit to have a level pathway between the computer room and the loading dock where the equipment will be delivered. Be sure to take into account any hallway corners to ensure that aisles are sufficiently wide to enable corners to be turned. Also, make note of sprinkler heads that will be below ceiling height on the path as well as door locking mechanisms and door jams on the floor that will reduce the effective clearance. Some equipment is sufficiently heavy that the use of metal plates is necessary to avoid floor damage or collapse during delivery to the computer room. With systems requiring much cooling, very large pipes carry very large volumes of water. These pipes may be under the floor or overhead. Smoke detectors and moisture detectors must be correctly installed. Most modern detection systems interface to a site management/security system. It is important to make sure the detection system is integrated so that the proper people are notified in a timely manner.

Power consideration begins with the ability of the utility company to deliver adequate power to the site from its substations. Be prepared for a shocked reaction from your utility company the first time you call and make your request, especially if you have never done this before. During installation, it is wise to label and record every path that the electrical supply will follow to enable quick traceback in the event of problems or electrical capacity questions.

## On-going non-personnel expenses

The power costs must not only take into account the power needs of the computer, but also the cost of the cooling. As a rule of thumb, multiply the power consumption of the system alone by 35-40% to estimate the additional power consumption of the required cooling. Today's rates for power vary substantially over the country, ranging from under 3 cents/kwh to over 10 cents/kwh.

First year maintenance may be included in the price of a new system. After that, unless the purchase has explicitly included multi-year maintenance, annual maintenance costs seem to range between 4-8% of the purchase price of the machine. It is not necessary to get a maintenance contract with extremely rapid response. For a system with a large node count, it is much more important to be able to remove a node from the system rapidly, reconfigure, preferably with spares, and continue. Next day service may be adequate for the vendor to then do any required hardware maintenance on the removed nodes. It is almost always better to negotiate maintenance options with the vendor while negotiating for the original system, for that is when you have most leverage with the vendor. It is wise to structure these as annual options so that you can cancel the maintenance contract with the vendor if you can find a better deal.

Operation expenses can be kept down by developing operator-free systems. For this, you need an extensive alerting infrastructure, which relays system events to system administrators via pagers or text messaging on their cell phones. Underlying it is a monitoring system extensive and reliable enough to report any of the anomalies that system operators would likely catch. You actually need a hierarchy of monitoring, from simple pass/fail on individual low level devices, like nodes, disks, etc. to high level testing of several components in sequence and verifying that the end-to-end results are correct.

As a new trend, the four to five year operating cost including maintenance, space, power, and cooling of a major computer, which for many years was a small part of the total cost of ownership of a system, is now becoming a much more significant factor, and may even exceed the original capital investment.

## Software

Increasingly, system software for debuggers, mathematical libraries, job scheduling, performance analysis, and even compilers, is provided by companies other than the hardware vendor. The cost of this required third party software can be substantial, and often the suppliers do not have early access to hardware from the vendors. Make certain that you understand exactly what software will be supplied with the system, and what arrangements the vendor has with the independent software vendors who will supply these other needed tools. The cost of these licenses can be large. However, it is not always necessary to license tools such as debuggers for the full system. For example, debugging tools are not very effective above 100-200 tasks, so don't bother to license the debugger for 2000 nodes. This can save a substantial amount of money. There are high-quality, robust mathematical libraries that are available

for free from universities and government laboratories as a result of many years of development from the NSF and DOE. Often, vendors have optimized versions of these libraries available for their systems.

## Systems and operations personnel tasks

There are a large number of different tasks that get lumped into systems and operations. We break them down into Core System Software, Machine Room Networking, User Access, Resilience, and Management. We briefly describe each of them, and return at the end to estimate the FTE effort required to carry them out.

**Core System Software** – This includes support for the *operating system* (OS), as well as for *tools* layered on top of the OS, including debuggers, scientific libraries, system monitoring displays, and many more. Get used to the idea that this work is never done. You will continually be installing new versions and patches. Best practices in version control are a necessity. New versions often introduce new bugs, and you will want the ability to fall back easily on the previous version. There are really two aspects to the OS and tool support. Some of it runs on the individual nodes of the system. Others are concerned with the system-wide aspects. Both need attention. Moreover, most HPC centers run multiple computing systems, each with a different OS, and each, of course, needs attention. For large systems, be prepared to have a system larger than what the vendor has for internal testing of software. This implies that patches and new system software versions may have never been tested on a large machine before being delivered. You should negotiate with the vendor to provide test time on your system to run validation and regression testing before installing new software.

Rather distinct are issues related to *file systems*. Often, there are at least three different file systems, which we can call home directories, local files, and system-wide files. Home directories are associated with individual users or projects. It is here that users store select information long term. These are usually backed up and often subject to quotas. Local file systems are local to the individual nodes, while system-wide file systems are globally accessible, and can often be written and read in parallel. Both sets of these files are viewed as temporary. They are associated with running jobs, or jobs which have recently run. Permanent file storage is found in the *mass store system*, which usually has a disk cache and a tape back end, with system algorithms that determine when to move the files from the rapidly accessible but expensive disk cache to the less rapidly accessible but much less expensive tape system. Data on the mass storage system is also usually backed up. Monitoring and managing the file systems is necessarily an ongoing operational requirement.

**Machine Room Networking** – The tasks include engineering support for the design, development, installation, operation, testing and debugging of all network infrastructure. At minimum, one needs a good background in network protocols, including but not limited to TCP/IP, as well as network diagnostics, end-to-end network performance and network routing.

**User Access** – The issues here can be categorized as user accounts, job processing, and reporting. *User accounts* have to be continually created, monitored and shut down. Authentication mechanisms have to be installed, which usually also means maintaining a Kerberos server. *Job processing* may include such things as developing and maintaining a scheduler, which implements scheduling policy, as well as exposing a queuing system that the users see. *Reporting* means processing individual job records and creating a database to easily answer questions such as how many users have used what resources in what discipline, what fraction of the usage has used what number of processors, how many new users have been added in the past year, and what is the demographics of the users. Today's management usually wants a web interface to be able to easily query the database and directly extract the kinds of information it needs.

Resilience means keeping the system as secure as possible, testing that it is operating reliably and taking the necessary steps if it is not. It includes such matters as security, monitoring system status, node management and regression testing. *Security* is an increasing concern at all large HPC sites. Items include reliable authentication of users, developing and enforcing best practices for staff including one-time passwords for those with system privileges. Knowing the availability of processors for jobs is not nearly as simple as it sounds. Usually it is felt that hardware errors should be dealt with by the manufacturer, while software errors can be cured by system reboot. But deciding whether errors are hardware or software is actually not clearcut. Nodes can seem to be available to the scheduler, but in fact are not. Most common is the case where processes from a previous job don't get cleaned up. Starting another job on this node will then not perform properly. *Node management and analysis* means both calling field service in case of hardware issues and maintaining an easily queried database to identify problematic nodes- those that fail repeatedly in hardware or software. Although most manufacturers support error logging, those logs often produce flat files and in many different places. Effort needs to be expended to collect that data into a useful database. Then the *data* has to be *analyzed*, so that one can spot problem nodes ahead of when they actually fail. Moreover, you cannot safely assume that nodes that have been returned by field service are either in good working order or even that the problem you had detected has been solved. After installation of new software or hardware, *regression testing* must re-establish that the system now gives acceptable answers and performance for all the tests in the test suite.

**Management** – This includes the staff time to manage the people, including hiring people, project management of carrying out the individual subtasks outlined above, management reporting, as well as keeping an up to date inventory of equipment. It also includes vendor relationships, not only with your current suppliers but with others as well, so that you can keep abreast of new technological developments that might be relevant for your next endeavors. For your current vendors, there is often significant effort in keeping track of the progress on problems you have reported that they have promised to fix.

## Personnel effort required

The FTE effort required to carry out the systems and operations functions clearly depends on many factors, such as the number of different systems, the size (processor count) of each, the number of users, whether or not you take early systems or wait until they are mature (i.e. others have worked out the inevitable bugs that are encountered in early systems). Estimates are that it takes 8-12 FTEs for a single, large, stand-alone HPC system to provide systems administration, security, networking, storage, and 24x7 monitoring and operational support. After that, when you add more people for each new system, the growth is much slower. However, Tim Thomas of the University of New Mexico reports (private communication) that he has looked at many systems and found the following amusing rule of thumb. For a small installation (less than 50 processors), you don't usually find a dedicated system person (this means that you are exploiting graduate students or the PI). After that, he finds that sites add one FTE for roughly every 250 processors. Of course, this rule shouldn't work because it is insensitive to the number of systems, the number of users, whether these are early systems or not. A quick look at other TeraGrid sites indicates that this rule gives a fairly good estimate of the systems effort expended. However, the more systems involved and the richer the underlying infrastructure, the more personnel are needed to provide stable support and assistance to users, so these numbers are highly dependent on the facility.

**Acknowledgements**
We received valuable input from Lynn Layman, J. Ray Scott, and Wendy Huntoon.

# Designing and Supporting Data Management and Preservation Infrastructure

Fran Berman
Reagan Moore
San Diego Supercomputer Center

## 1. Introduction

The 20th century brought about an "information revolution" that has forever altered the way we work, communicate, and live. In the 21st century, data is ubiquitous. Available in digital format via the web, desktop, personal device, and other venues, data collections both directly and indirectly enable a tremendous number of advances in modern science and engineering.

Today's data collections span the spectrum in discipline, usage characteristics, size, and purpose. The life science community utilizes the continually expanding *Protein Data Bank*[1] as a worldwide resource for studying the structures of biological macromolecules and their relationships to sequence, function, and disease. The *Panel Study of Income Dynamics* (PSID),[2] a longitudinal study initiated in 1968, provides social scientists detailed information about more than 65,000 individuals spanning as many as 36 years of their lives. The *National Virtual Observatory*[3] is providing an unprecedented resource for aggregating and integrating data from a wide variety of astronomical catalogs, observation logs, image archives, and other resources for astronomers and the general public. Such collections have broad impact, are used by tens of thousands of individuals on a regular basis, and constitute critical and valuable community resources.

However, the collection, management, distribution, and preservation of such digital resources does not come without cost. Curation of digital data requires real support in the form of hardware infrastructure, software infrastructure, expertise, human infrastructure, and funding. In this article, we look beyond digital data to its supporting infrastructure, and provide a holistic view of the software, hardware, human infrastructure, and costs required to support modern data-oriented applications in research, education, and practice.

## 2. Digital Data Curation

Digital *data curation* focuses on the generation of descriptive metadata and validation of the quality of the data. Digital *data preservation* focuses on the characterization of the data authenticity (provenance information), and the management of data integrity across multiple generations of storage technology. An example of a curated community digital collection is the *Protein Data Bank* (PDB). The PDB is a global resource for structural information about proteins that is maintained by the Worldwide Protein Data Bank (wwPDB). This organization is composed of the Research Collaboratory for Structural Bioinformatics (RCSB), a consortium consisting of groups at UCSD/SDSC and Rutgers; the Macromolecular Structure Database (MSD) at the European Bioinformatics Institute (EBI) in Hinxton, UK; and PDBj in Osaka Japan.

When a user accesses a data portal for the PDB for information on HIV-1 protease, a target in the fight against AIDS, considerable infrastructure is provided to support this. Behind the scenes, the following components are involved in providing information on HIV-1 protease to the user:

[1] http://www.pdb.org/pdb/Welcome.do

[2] http://psidonline.isr.umich.edu/

[3] http://www.us-vo.org/

- Data are collected, annotated and validated at one of three wwPDB deposition sites; in the US this site is located at the Rutgers site of RCSB. The wwPDB has adopted the PDB Exchange Dictionary as a means of standardizing semantics to ensure uniform data and provides the foundation for data exchange within the wwPDB and delivery of a standard data representation to the public.
- The acquisition, annotation and validation of PDB data requires about 20 highly trained personnel as well as significant computational, storage and network resources. An individual PDB dataset may consist of more than 1000 individual data items with some containing as many as 1M instances. Data acquisition requires reliable, low-latency network connections and high-performance servers to deliver real-time data validation. The validation and computation of derived features for these datasets is computationally intensive and is performed on clusters of Linux and Solaris servers at each of the wwPDB sites. PDB data that are ready for public release are transferred to the RCSB PDB site at SDSC, the main US PDB distribution site.
- The RCSB PDB data portal accessed by the user is served from a Linux Cluster at SDSC using software written and maintained by the RCSB PDB team in collaboration with SDSC researchers. The portal comprises a Linux commodity cluster controlled by dual Cisco load balancers that handle traffic from 10,000 scientists a day making 2M page requests. Each cluster node has a redundant copy of the PDB (approx 1TB) and state is maintained using JBOSS and the load balancers. A failover is provided using UltraDNS (third party) that fails over to a small cluster at Rutgers University in New Jersey on the rare occasions that power or networking is lost to all nodes at SDSC. In this way 99.99% uptime is maintained. The RCSB PDB previously maintained mirrors around the world for fast access, but current high-speed networking to SDSC is good enough that these are no longer needed for this purpose.
- A group of nine staff and students in the PDB group at UCSD develop access and mining tools for the PDB community. In addition to the PDB site at UCSD the wwPDB sites in Europe and Japan provide complementary views of the common set of archival data.
- The PDB currently requires 20TB of storage associated with the collection at SDSC alone.
- Additional costs for maintaining the curation involve business office and HR costs, the costs of administering the project by its leadership and management, the costs of advisory and evaluation input (travel, time, etc.), and other related costs.

The RCSB PDB database infrastructure at UCSD is coded in Java and completely built from public domain software. A MySQL database is used at SDSC to instantiate this schema, with middle and presentation layers built around Hibernate. Focus on software development includes systems to organize the material and provide services to support discovery, browsing, and presentation. The RCSB PDB infrastructure was developed to allow the collection to expand continuously and to allow ingestion and evaluation for new entries.

In short, to accommodate the request for information about HIV-1 protease from the PDB, substantial software, hardware, human, and funding support is required. At a recent AAAS Panel on data,[4] RCSB PDB Director Helen Berman estimated that in 2005 more than one billion dollars of research funding were spent to generate the data that were collected, curated and distributed by the PDB.

[4] http://php.aaas.org/meetings/MPE_01.php?detail=1110

## 3. Preserving Data over Time

Some digital collections will continue to be valuable resources for the foreseeable future. These typically include irreplaceable collections (e.g., the Shoah Collection of Holocaust survivor testimony),[5] valuable community reference collections (e.g., PDB, NVO, PSID), and historically valuable collections such as federal digital records.[6] [7] For these digital collections, lifetime is measured in decades, with continuous active preservation, and often new material is added over time. Over a collection's decades of existence, the media on which it is stored will go through tens of generations, standard encoding formats will evolve, preservation staff and institutions may change, etc. In short, everything involved with the collection may evolve, and evolution must be planned and executed in a way that maintains the integrity of the data collection and minimizes disruption to access from its user community.

[5] http://www.usc.edu/schools/college/vhi/

[6] http://www.archives.gov/

[7] http://www.loc.gov/index.html

Because the time periods over which long-term digital collections are preserved are measured in decades, the need for preservation environments is critical. At SDSC, some of the current data collections have been migrated over the last 20 years onto six generations of storage technology. Over that period, the trend in tape media costs per byte has been exponential, dropping by half approximately every three years. If this exponential trend continues, the total life-time cost of media is only twice the original media cost, being

$$(1 + 1/2 + 1/4 + \ldots) * (\text{original cost}).$$

Of course, tape media are only a modest portion of the true cost of long-term storage and the labor for administering the storage system, in particular managing the transitions between generations of storage technology, must be incorporated into cost models (see below). Generally, the number of individuals managing the collections can stay constant, after the initial period of implementation, even though both the size of the data files and the size of the storage media are growing. This means that costs related to storage management labor are increasing slower than costs related to collection building and maintenance.

## 4. Data Management and Preservation for the Science and Engineering Community

Today's large-scale computational runs often result in large-scale data output. It is not uncommon for a simulation to generate a million files and tens of terabytes of data with over 30 individuals collaborating on the application runs. This level of data output requires dedicated handling to move the data from the originating disk cache into a digital library for future access, with replication on an archival storage system.

SDSC's digital data collections are representative of the state of the art. Digital collections developed for specific scientific disciplines typically have unique usage models but can share the same evolving data management infrastructure, with the difference between usage and storage models mainly tied to differences in management policies for sustainability and governance. Table 1 lists three categories of digital holdings at SDSC, loosely characterized as *data grids* (primarily created to support data sharing), *digital libraries* (created to formally publish the digital holdings), and *persistent archives* (focused on the management of technology evolution).

Data management requirements can be derived from Table 1. Today, it is not uncommon for a collection to contain 10 to 100 hundred terabytes of data, with two to 10 million files. In

fact, collections are now assembled that have too many files to house in a single file system – containers are used to aggregate files into a larger package before storage, or files are distributed across multiple file systems. The number of individuals that collaborate on developing a shared collection can range from tens to hundreds. In Table 1, the column on the right labeled ACLs (Users with Access Controls) shows how many individuals (including staff) are typically involved in writing files, adding metadata, or changing the digital holdings in the collection. The number of individuals who access the collection can be much larger, as most of the collections are publicly accessible.

| Date | | 5/17/02 | | 6/30/04 | | | 1/3/06 | | |
|---|---|---|---|---|---|---|---|---|---|
| Project | GBs of data stored | 1000's of files | GBs of data stored | 1000's of files | Users with ACLs | GBs of data stored | 1000's of files | Users with ACLs |
| Data Grid | | | | | | | | |
| NSF / NVO | 17,800 | 5,139 | 51,380 | 8,690 | 80 | 93,252 | 11,189 | 100 |
| NSF / NPACI | 1,972 | 1,083 | 17,578 | 4,694 | 380 | 34,452 | 7,235 | 380 |
| Hayden | 6,800 | 41 | 7,201 | 113 | 178 | 8,013 | 161 | 227 |
| Pzone | 438 | 31 | 812 | 47 | 49 | 19,674 | 10,627 | 68 |
| NSF / LDAS-SALK | 239 | 1 | 4,562 | 16 | 66 | 104,494 | 131 | 67 |
| NSF / SLAC-JCSG | 514 | 77 | 4,317 | 563 | 47 | 15,703 | 1,666 | 55 |
| NSF / TeraGrid | | | 80,354 | 685 | 2,962 | 195,012 | 4,071 | 3,267 |
| NIH / BIRN | | | 5,416 | 3,366 | 148 | 13,597 | 13,329 | 351 |
| Digital Library | | | | | | | | |
| NSF / LTER | 158 | 3 | 233 | 6 | 35 | 236 | 34 | 36 |
| NSF / Portal | 33 | 5 | 1,745 | 48 | 384 | 2,620 | 53 | 460 |
| NIH / AfCS | 27 | 4 | 462 | 49 | 21 | 733 | 94 | 21 |
| NSF / SIO Explorer | 19 | 1 | 1,734 | 601 | 27 | 2,452 | 1,068 | 27 |
| NSF / SCEC | | | 15,246 | 1,737 | 52 | 153,159 | 3,229 | 73 |
| Persistent Archive | | | | | | | | |
| NARA | 7 | 2 | 63 | 81 | 58 | 2,703 | 1,906 | 58 |
| NSF / NSDL | | | 2,785 | 20,054 | 119 | 5,205 | 50,586 | 136 |
| UCSD Libraries | | | 127 | 202 | 29 | 190 | 208 | 29 |
| NHPRC / PAT | | | | | | 101 | 474 | 28 |
| TOTAL | 28 TB | 6 mil | 194 TB | 40 mil | 4,635 | 655 TB | 106 mil | 5,383 |

Table 1. Evolution of digital holdings at SDSC

For many digital holdings, the collection may be replicated among different storage systems and/or sites. The replication serves multiple purposes:

- *To meet governance and sustainability policies*, with a copy at the institution that has assumed long-term management of the collection
- *To mitigate the risk of data loss.* At least five different loss mechanisms are mitigated through replication; media corruption (e.g., disk crash or tape parity error), systemic vendor product error (such as bad microcode in a tape drive), operational error, malicious user attack, and natural disaster (e.g., fire, flood, hurricane, etc.).
- *To improve access via disk caches.* Wide-area-networks are characterized by access latencies (typically tens to hundreds of milliseconds) that are substantially higher than that of a spinning

disk. Replicating data onto a local disk cache ensures interactive access for local users. Replicating data onto a remote disk cache ensures interactive access for the remote users.

- *To provide high availability.* Having multiple independent copies ensures that when any single system component is taken offline for maintenance, or is down because of failure, the digital holdings can still be accessed.

For many collections, data sources are inherently distributed. The National Virtual Observatory collection provides an example of this. Thus, a data management environment must provide the capabilities needed to manage data distributed over a wide-area-network. This requirement can be characterized as *latency management* and is typically achieved by minimizing the number of messages that are sent over wide-area-networks. Common mechanisms for latency management include

- replication,
- bulk operations for manipulating small files and loading metadata, and
- remote procedures to parse or filter data directly at the remote storage system.

Many data collections at SDSC are managed on top of federated data grids. Having multiple independent data grids, each with a copy of the data and metadata (both descriptive attributes and state information generated by operations on the data), ensures that no single disaster can destroy the aggregated digital holdings. Federation allows the management of shared name spaces between the independent data grids, enabling the cross registration of files, metadata, user names, and storage resources. The types of federation environments range from *peer-to-peer data grids*, with only public information shared between data grids, to *central archives* that hold a copy of records from otherwise independent data grids, to *worker data grids* that receive their data from a master data grid.

## 5. Data Management System Components and Costs

The data management systems supporting digital collections support a variety of functions including sharing of data, publishing of data, preserving of data, and analyzing of data.

Data management system components may include the following:

- *Authenticity system for validating the identity of users*
- *Authorization software system* for controlling updates
- *Disk cache* for interactive access
- *Archival storage* for long term preservation of data
- *Databases* for organizing metadata for each collection
- *Data grids* for managing distributed data
- *High-performance networks* between the disk caches and archives, capable of supporting parallel I/O streams
- *Workflow software* for managing curation and preservation processes such as checksum validation, synchronization of replicas, transformative migration of encoding formats to new standards.

- *Portals* for managing access to the collections, including digital library services for presenting the data
- *Analysis systems* for applying classification and categorization filters to the data
- *Knowledge systems* for managing the resulting relationships or inferences that have been made on the data

The **costs** of such data management environments are driven by the need for integrity (eg., multiple replicas, validation of checksums over time, management of access controls), authenticity (management of provenance information to understand data context), scalability (management of the future number of files and amount of storage), and access (support for interactive versus batch access). By minimizing capabilities, the cost can be reduced. For a system that promotes the advancement of science, ability to support intensive analysis is key. For a system that ensures high reliability, the risk of data loss must be minimized.

**Component costs of the data management system** include the costs of installation, maintenance, and evolution of

- **Authentication and authorization systems.** Data grids are able to use the Grid Security Infrastructure to authenticate users. This requires a Certificate Authority and an associated dedicated server.
- **Disk storage.** Today, disk storage costs between $1000-$5000 per Terabyte per year (amortized cost of capital equipment and labor to administer the storage) depending on the type of disk and built-in redundancy (e.g., mirroring).
- **Tape storage.** Current tape archives cost about $400 per Terabyte per year (amortized cost of media, tape silos, archive software and labor to administer the storage). The amortized media cost is about 1/6 of this annual cost. However, note that this is the cost for a single copy. Three copies are preferred to minimize risk of data loss (original plus two replicas). If one of these copies is kept on disk, then only two tape copies are needed, preferably stored at different locations on different vendor equipment under different administrative control.
- **The Database.** The metadata used for provenance and discovery is stored on-line in a database. A single database can support multiple database instances, allowing all of the collections to be managed using the same software. The cost of management of a database instance is about $5000 per year, for a database that supports 15-20 instances. The cost of the database software depends upon the vendor, with open-source databases requiring more local expertise to run and commercial databases requiring a service contract.
- **The Data grid.** The software that supports distributed data is freely available to academic institutions, and the administrative support is similar to that of the database administrator.
- **The High-performance network.** Note that the movement of a Terabyte of data per day is equivalent to a sustained data rate of 11.6 MB/second. Current collections at SDSC are growing in size at the rate of 1-2 Terabytes per day. The replication and access of this data requires networks that sustain 25-50 MB/second.
- **Workflow software.** The platforms that support the workflows in the past have been the same as the application computer platforms and the data analyses. For the manipulation and analysis of 10 Terabytes of data per day, 10-Teraflop systems are required.
- **User portals.** The server supporting the portal or digital library interface is typically accessed over the web. This implies the need to support JSR 168 java portlets as well as web servers. This is typically done on a server separate from the database server.

- **The Knowledge system.** The management of relationships on the data is enabled by modern digital library middleware. This system typically runs on the same server as the database technology.

## 6. Management

The long-term management of data requires a sustainability and governance model that specifies the policies that will be used to guarantee funding support, minimize risk of data loss, assure integrity, and assure authenticity.

The management plan needs to address plans for future access if the sustainability model fails, where the collection might be housed, and how the material will be migrated to the new environment. The concept of infrastructure independence in persistent archives can be extended to include independence from a particular sustainability model through federation with other institutions that use alternate sustainability models. Guaranteed access to a collection requires a community that is willing to curate the collection, identify risks to the maintenance of the collection, and seek opportunities to replicate the collection as widely as possible.

## 7. Conclusion

For science and engineering, as in life, there is "no free lunch." The ability to organize, analyze, and utilize today's deluge of data to drive research, education, and practice incurs costs for management, curation, preservation and distribution. These costs must be included in project budgeting and infrastructure planning, and are non-zero.

They are better than the alternative, however. Without responsible data planning as part of the process of project development, organization, and management, valuable data collections will be lost, damaged, or become unavailable. Lack of planning can incur substantive cost for resurrecting, re-generating, or rescuing a data collection, and without critical data, science and engineering advancement and discovery can be slowed. At the end of the day, the costs of thoughtful and strategic data management, curation and preservation are a bargain.

**Acknowledgements**
The authors would like to thank Helen Berman, Phil Bourne, and Richard Moore for their comments and improvements.

# NWChem
## Development of a Modern Quantum Chemistry Program

Thom H. Dunning, Jr
National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign

Robert J. Harrison
Jeffrey A. Nichols
Computing and Computational Sciences Directorate, Oak Ridge National Laboratory

### 1. Background

In the 1980s, it became clear that decommissioning and rehabilitation of the nuclear weapons complex operated by contractors of the U.S. Department of Energy (DOE) was a monumental challenge. The weapons sites contained tens of millions of gallons of high level radioactive wastes and hundreds of cubic kilometers of contaminated soils as well as thousands of contaminated facilities. Towards the end of the 1980s, Robert S. Marianelli, Director of the Chemical Sciences Division in DOE's Office of Science (DOE-SC) and William R. Wiley, Director of the Pacific Northwest National Laboratory began laying plans for a major new laboratory that would focus on gaining the fundamental understanding needed to tackle these problems. Their work eventually led to the construction of the Environmental Molecular Sciences Laboratory (EMSL), a national user facility dedicated to molecular research related to environmental science and waste processing.

The size of molecular systems involved in environmental science (*e.g.*, aqueous solutions) and high level wastes (*e,g.*, trans-uranic compounds and metal ion chelating agents) was considerably beyond those that could be studied with the molecular modeling software and computing resources available at the time. A workshop was convened in February 1990 to discuss the approach to be taken. The report from the workshop recommended that the DOE-SC establish a major new computing facility in the EMSL *and*, simultaneously, make a major investment in the development of new quantum chemistry software designed explicitly for massively parallel computing systems. Thus began the development of the Northwest Chemistry package (NWChem).[1][2][3][4] Although the official start of the project would be delayed for another couple of years, work began soon thereafter exploring technologies that could be used for a new, scalable quantum chemistry application that included the major atomic and molecular electronic structure methods (*e.g.*, Hartree-Fock, perturbation theory, coupled cluster theory, etc.) as well as molecular dynamics simulations with empirical, semiempirical or *ab initio* potentials.

One of the authors (Dunning) instigated the NWChem project, while the other authors were the chief architect (Harrison) and project manager (Nichols).

### 2. Development of NWChem

In the early 1990s, the development of quantum chemistry software that would scale to hundreds, if not thousands, of processors was a challenging task. It was not known how to parallelize some of the basic mathematical algorithms used in molecular computations, let alone how to parallelize the many algorithms specific to these computations. In addition, many basic software technologies (*e.g.*, interprocessor communication) were still evolving. To address these issues, the NWChem Project Team included theoretical and computational chemists, computer scientists, and applied mathematicians. This long-term partnership, which continues to this day, was critical to meeting the goals of the NWChem Project and led to the development of new mathematical algorithms (*e.g.*, PEIGS for diagonalizing matrices) as well as to new computer system technologies (*e.g.,* Global Arrays for interprocessor communication). The multidisciplinary NWChem Project served as a model for the approach taken in DOE-SC's Scientific Discovery through Advanced Computing (SciDAC) program.

[1] Guest, M.F.,Apra, E., Bernholdt, D.E., Fruechtl, H.A., Harrison, R.J., Kendall, R.A., Kutteh, R.A., Long, X., Nicholas, J.B., Nichols, J.A., Taylor, H.L., Wong, A.T., Fann, G.I., Littlefield, R.J., Nieplocha, J. "High Performance Computational Chemistry; NWChem and Fully Distributed Parallel Algorithms," High Performance Computing: Issues, Methods, and Applications. Eds. Dongarra, J., Gradinetti, L., Joubert, G., Kowalik, J.: 1995.

[2] Kendall, R.A., Apra, E., Bernholdt, D.E., Bylaska, E.J., Dupuis, M., Fann, G.I., Harrison, .J., Ju, J., Nichols, J.A., Nieplocha, J., Straatsma, T.P., Windus, T.L., Wong, A.T. "High performance computational chemistry: An overview of NWChem a distributed parallel application," Comput. Phys. Commun. 128 (2000): pp. 260.

[3] For a brief period very early in the project the name BATMOL (due to J. Anchell) was used. Species of northwest salmon (chinook, king, coho, etc.) were also considered as names for the code. NWChem was adopted as the name to advertise its institutional origin and science goals. Finally, NWChem is pronounced "N-W-Chem" – the forms "new-chem" and "nuke-em" are incorrect.

[4] http://www.emsl.pnl.gov/docs/nwchem/nwchem.html

The NWChem project was supported by the Office of Biological and Environmental Research (OBER) in DOE-SC as an integral part of the EMSL Project. The EMSL Project provided approximately $2 million per year for the period FY1992-7. During this same period, DOE-SC's Office of Advanced Scientific Computing Research provided another $0.5 million per year to support a Grand Challenge project in computational chemistry. The NWChem project leveraged many of the results from the Grand Challenge project. In addition, the initial exploratory work was funded by the Laboratory Director's Research and Development program at the Pacific Northwest Laboratory. The "core" NWChem project team involved five computational chemists and three computer scientists and applied mathematicians. In addition, 14 postdoctoral fellows were involved in the project. All told, it is estimated that approximately 100 person-years and $12 million were devoted to the development of NWChem v1.0, not including the effort required to develop the technology incorporated in NWChem from external sources.

Several goals were set for the NWChem software package. These included:

1. NWChem would be based on algorithms that:
   - scale to hundreds, if not thousands, of processors, and
   - use the minimum number of flops consistent with the above.
2. NWChem would be affordable to develop, maintain, and extend.
3. NWChem would be independent of computer architecture to the maximum extent possible:
   - a hardware abstraction layer would be used to isolate the most of the program from the details of the hardware, but
   - it would be possible to tune the code to some extent to optimize performance on the specific computer being used.

Achieving these goals required a combination of research to determine the best solutions to the above problems, modern software engineering practices to implement these solutions, and a worldwide set of collaborators to provide expertise and experience missing in the core NWChem software development team. Fifteen external collaborators were involved in the development of NWChem; seven from the US, seven from Europe, and one from Australia.

A number of basic computing issues had to be addressed to optimize the performance and scalability of NWChem. These included: processor architecture, node memory latency and bandwidth, interprocessor communications latency and bandwidth, and load balancing. Solving the associated problems often required rewriting and restructuring the software, explorations that were carried out by the postdoctoral fellows associated with the NWChem project. Another issue that was always in the foreground was the portability of the software. Computational chemists typically have access to a wide range of computer hardware, from various brands of desktop workstations, to various brands of departmental computers, to some of the world's largest supercomputers. To most effectively support their work, it was important that NWChem run on all of these machine, if possible.

The process for designing, developing and implementing NWChem used modern software engineering practices. The process can be summarized as follows:

1. *Requirements gathering.* The process began by gathering requirements from the researchers associated with the EMSL Project. This defined the functionality that had to be provided by the quantum chemistry software.

2. *Preliminary design and prototyping.* After the requirements were gathered, work on NWChem began. This included design of the overall system architecture, identification of the major subsystems, definition of the objects and modules, definition of the internal and external interfaces, characterization of the major algorithms, etc.
3. *Resolution of unresolved issues.* The preliminary design work led to the identification of a number of major, unresolved issues. Research projects were targeted at each of these issues.
4. *Detailed design*. In the meantime, the preliminary design was extended to a set of "code to" specifications. As the major issues were resolved, they were included in the "code to" specifications.
5. *Implementation.* NWChem was then created in well defined versions and revision control was used to track the changes.
6. *Testing and Acceptance.* Finally, a bevy of test routines were used to verify the code and ensure that the requirements were met.

Although the above is a far more rigorous process that is followed in most scientific software development projects, we found it to be critical to meeting the goals set for NWChem and for managing a distributed software development effort. The above cycle was actually performed at least twice for each type of NWChem method implemented (*e.g.*, classical, uncorrelated quantum, highly correlated quantum, density functional, etc). Going through the cycle multiple times generated "beta" software that could be released to users for feedback and refinement of user requirements.

Although the combination of an on-site core team plus off-site collaborators provided the range of technical capabilities needed to develop NWChem, there are lessons to be learned about managing such a highly distributed project. For example

- The time and effort required for integration of existing sequential or parallel codes into the new code framework was always larger than estimated.
- The preparation of documentation, for both users and programmers, should have been initiated earlier in the project. The programmer's manual is especially important because this document provides the guidelines needed to ensure that the software produced by the distributed team will work together.
- Software components that are on the critical path should be developed in-house, since the time schedules and priorities of collaborators inevitably differ from those of the core team.
- It is important to implement code reviews both for software developed in-house by the "core" team as well as that developed by the external collaborators.

Our experience suggests that a distributed software development team can be successful *if* the core team is large enough to develop all of the software components on the critical path and *if* sufficient guidance is provided to the collaborators on the format and content for their contributions and their progress is carefully monitored.

## 3. Architecture of NWChem

In addition to achieving high performance and being scalable to large numbers of processors, scientific codes must be carefully designed so that they can easily accommodate new mathematical models and algorithms as knowledge advances. If scientific codes can not evolve as new knowledge

is gained, they will rapidly become outdated. It must also be possible to move the codes from one generation of computers to the next without undue difficulty as computer technology advances—the lifetime of scientific codes is measured in decades, the lifetime of computers in years.

For the above reasons, NWChem is best thought of as a framework or environment for chemical computation rather than a single, fully integrated application. The framework defines and supports a "virtual machine" model and mandates a certain structure for new modules. A well-defined virtual machine model hides details of the underlying hardware and encourages programmers to focus on the essentials; correctness, good sequential performance, expressing concurrency, and minimizing data motion. The high-level structure ensures correct operation, provides a consistent look-and-feel for users, and enables code reuse. New chemical functionality can then be developed using a well defined set of capabilities, which are described below. Testimony to the success of this framework (and perhaps to the inadequacy of our current programmer's manual) is a recent comment from a developer new to NWChem – "my program is running correctly in parallel but I don't know how."

The key to achieving the above goals is a carefully designed architecture that emphasizes layering and modularity (Fig. 1). At each layer of NWChem, subroutine interfaces or styles were specified in order to control critical characteristics of the code, such as ease of restart, the interaction between different tasks in the same job, and reliable parallel execution. Object-oriented design concepts were used extensively within NWChem. Basis sets, molecular geometries, chunks of dynamically allocated local memory, and shared parallel arrays are all examples of "objects" within NWChem. NWChem is implemented in a mixture of C and Fortran-77, since neither C++ nor Fortran-90/95 were suitable at the start of the project. Since we did not employ a true object-oriented language, and in particular did not support inheritance, NWChem does not have "objects" in the strict sense of the word. However, careful design with consideration of both the data and the actions performed upon the data, and the use of data hiding and abstraction, permits us to realize many of the benefits of an object-oriented design.

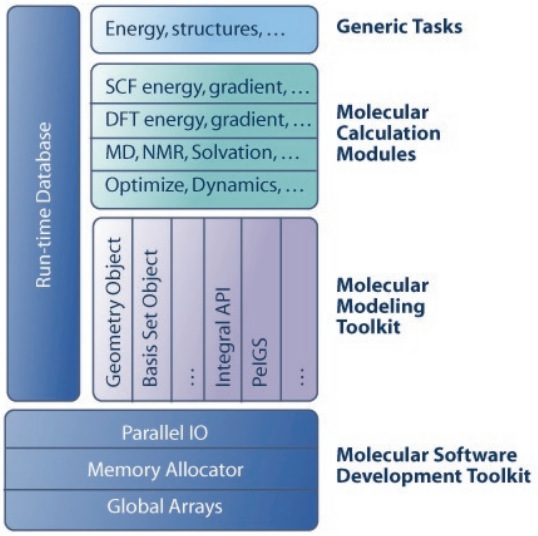


Figure 1. Layered, modular architecture of NWChem.

In the very bottom layer of NWChem is the Software Development Toolkit. It includes the Memory Allocator (MA), Global Arrays (GA), and Parallel IO (ParIO). The "Software Development Toolkit" was (and still is) the responsibility of the computer scientists involved in NWChem. It essentially defines a "hardware abstraction" layer that provides a machine-independent interface to the upper layers of NWChem. When NWChem is ported from one computer system to another, nearly all changes occur in this layer, with most of the changes elsewhere being for tuning or to accommodate machine specific problems such as compiler flaws. The "Software Development Toolkit" contains only a small fraction of the code in NWChem, less than 2%, and only a small fraction of the code in the Toolkit is machine dependent (notably the address-translation and transport mechanisms for the one-sided memory operations).

The next layer, the "Molecular Modeling Toolkit," provides the functionality commonly required by computational chemistry algorithms. This functionality is provided through "objects" and application programmer interfaces (APIs). Examples of objects include basis sets and geometries. Examples of the APIs include those for the integrals, quadratures, and a number of basic mathematical routines (*e.g.*, linear algebra and Fock-matrix construction). Nearly everything that might be used by more than one type of computational method is exposed through a subroutine interface.

# NWChem
## Development of a Modern Quantum Chemistry Program

Common blocks are not used for passing data across APIs, but are used to support data hiding behind APIs.

The runtime database (RTDB) is a key component of NWChem, tying together all of the layers of NWChem. Arrays of typed data are stored in the database using simple ASCI strings for keys (or names) and the database may be accessed either sequentially or in parallel.

The next layer within NWChem, the "Molecular Calculation Modules," is comprised of independent modules that communicate with other modules only via the RTDB or other persistent forms of information. This design ensures that, when a module completes, all persistent information is in a consistent state. Some of the inputs and outputs of modules (via the database) are also prescribed. Thus, all modules that compute an energy store it in a consistently named database entry—in this case <module>:energy, substituting the name of the module for <module>. Examples of modules include computation of the energy for SCF, DFT, and MCSCF wave functions. Surprising to some, the code to read the user input is also a module. This makes the behavior of the code more predictable, e.g., when restarting a job with multiple tasks or steps, by forcing the state of persistent information to be consistent with the input already processed. Modules often invoke other modules.

The highest layer within NWChem is the "task" layer, sometimes called the "generic-task" layer. Functions at this level are also modules—all of their inputs and outputs are communicated via the RTDB, and they have prescribed inputs and outputs. However, these capabilities are no longer tied to specific types of wave functions or other computational details. Thus, regardless of the type of wave function requested by the user, the energy may always be computed by invoking task_energy() and retrieving the energy from the database entry named task:energy. This greatly simplifies the use of generic capabilities such as optimization, numeric differentiation of energies or gradients, and molecular dynamics. It is the responsibility of the "task"-layer routines to determine the appropriate module to invoke.

NWChem was designed to be extensible in several senses. First, the clearly defined task and module layers make it easy to add substantial new capabilities to NWChem. Second, the wide selection of lower-level APIs makes it easier to develop new capabilities within NWChem than within codes in which these capabilities are not easy to access. Finally, having a standard API means that a change to an implementation will affect the whole code.

*Virtual Machine Model – Non-uniform Memory Access (NUMA)*

By the late 1980s it was apparent that distributed-memory computers were the only path to truly scalable computational power and the only portable programming model available for these systems was message passing. Although NWChem initially adopted the TCGMSG message passing interface, members of the NWChem team participated in development of the message passing interface (MPI) standard,[5] and the official NWChem message-passing interface has been MPI for several years. Without fear of contradiction, the MPI standard has been the most significant advancement in practical parallel programming in over a decade, and it is the foundation of the vast majority of modern parallel programs. The vision and tireless efforts of those who initiated and led this communal effort must be acknowledged. It has also been pointed out that the existence of such a standard was a prerequisite to the emergence of very successful application frameworks such as PETSc.[6]

A completely consistent (and deliberately provocative) viewpoint is that MPI is evil. The emergence of MPI coincided with an almost complete cessation of parallel programming tool/

[5] Dongarra, J., et al. "Special issue – MPI – a message-passing interface standard," Int. J. Supercomputer Appl. and High Perf. Comp. 8 (1994) : pp. 165.

[6] http://ww-unix.mcs.anl.gov/petsc/petsc-as/

paradigm research. This was due to many factors, but in particular to the very public and very expensive failure of HPF. The downsides of MPI are that it standardized (in order to be successful itself) only the primitive and already old communicating sequential process[7] (CSP) programming model, and MPI's success further stifled adoption of advanced parallel programming techniques since any new method was by definition not going to be as portable. Since one of the major goals of NWChem was to enable calculations larger than would fit into a single processor, it was essential to manage distributed data structures. Scalable algorithms also demand dynamic load balancing to accommodate the very problem dependent sparsity in matrix elements and wide ranging cost of evaluating integrals. Both of these tasks are difficult to accomplish using only simple message passing and a more powerful solution was demanded.

The Global Arrays (GA) toolkit[8 9 10] provides an efficient and portable "shared-memory" programming interface for distributed-memory computers. Each process in a MIMD parallel program can asynchronously access logical blocks of physically distributed, dense multi-dimensional arrays, without need for explicit cooperation by other processes (Fig. 2). Unlike other shared-memory environments, the GA model exposes to the programmer the non-uniform memory access (NUMA) characteristics of the high performance computers and acknowledges that access to a remote portion of the shared data is slower than to the local portion. Locality information for the shared data is available, and direct access to local portions of shared data is provided. The GA toolkit has been in the public domain since 1994 and is fully compatible with MPI.

Essentially all chemistry functionality within NWChem is written using GA. MPI is only employed in those sections of code that benefit from the weak synchronization implied by passing messages between processes, for instance to handle the task dependencies in classical linear algebra routines or to coordinate data flow in a highly optimized parallel fast Fourier transform. This success is due to combining the correct abstraction (multi-dimensional arrays of distributed data) with the programming ease and scalability of one-sided access to remote data. Performance comes from algorithms (Fig. 3) designed to accommodate the NUMA machine characteristics, e.g., Hartree-Fock,[11] four-index transformation,[12] and multi-reference CI.[13]

[7] Hoare, C.A.R "Communicating sequential processes," Communications of the ACM. 21 (8) 1978: pp. 666-677.

[8] http://ww.emsl.pnl.gov/docs/global/

[9] Nieplocha, J., Harrison, R.J., Littlefield, R.J. "Global Arrays: A nonuniform memory access programming model for high-performance computers," The Journal of Supercomputing, 10 (1996): pp. 197-220.

[10] Nieplocha, J., Harrison, R.J., Krishnan, M., Palmer, B., Tipparaju, V. "Combining shared and distributed memory models: Evolution and recent advancements of the Global Array Toolkit," Proceedings of. POOHL'2002 workshop of ICS-2002, NYC, 2002.

[11] Harrison, R.J., Guest, M.F., Kendall, R.A., Bernholdt, D.E., Wong, A.T., Stave, M., Anchell, J.L., Hess, A.C., Littlefield, R.J., Fann, G.I., Nieplocha, J., Thomas, G.S., Elwood, D., Tilson, J.L., Shepard, R.L., Wagner, A.F., Foster, I.T., Lusk E, Stevens, R. "Toward high-performance computational chemistry: II. A scalable self-consistent field program," J. Comput. Chem. 17 (1996): pp. 124.

[12] Wong, A.T., Harrison, R.J., Rendell, A.P. "Parallel direct four-index transformations," Theor. Chim. Acta 93 (1996): pp. 317.

[13] Dachsel, H., Nieplocha, J., Harrison, R.J. "An out-of-core implementation of the COLUMBUS massively-parallel multireference configuration interaction program," Proceedings of Supercomputing'98. 1998: pp. 41.

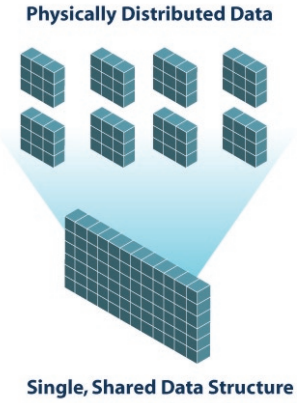


Figure 2. Global Arrays: each process in an MIMD parallel program can asynchronously access logical blocks of physically distributed, dense, multi-dimensional arrays, without need for explicit cooperation by other processes.

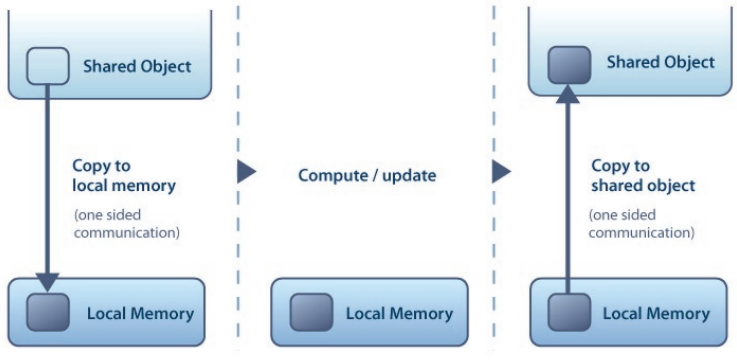


Figure 3. Non-uniform memory access (NUMA) model of computation. Each process independently moves data from a shared data structure to local memory for computation. Results can be written or accumulated into another shared structure. A simple performance model can be used to ensure that the cost of moving data is offset by the amount of computation performed.

# NWChem
## Development of a Modern Quantum Chemistry Program

## 4. Maintenance and Evolution of NWChem

When discussing computer software, it is impossible to ignore the on-going cost associated with maintenance and evolution of the software. This phenomenon is well understood by anyone who uses computers in their daily life. Even for our personal computers, there is a continuing level of effort associated with fixing bugs in, as well as adding new features to, the software. The need for maintenance and evolution of scientific and engineering software is several-fold greater. This is a result of the extraordinary complexity of the software coupled with the continual development of new methods resulting from increased scientific understanding and the technological upheaval associated with the often rapid evolution of cutting-edge computing technologies.

In the early formative years of NWChem development prior to the first official release of version 1.0 in 1997, it was critical to get the software out to users as quickly as possible, as often as possible, and always "for free." This allowed feedback from the users, including bug reports and fixes, as well as established a large user base. In addition, early usage generated revised user requirements, *e.g.*, a very important element of computational chemistry (Density Functional Theory) was added to the development effort (in 1993) after the project had already been initiated. The first beta releases of NWChem occurred in 1994 with subsequent trial releases occurring annually until the first official release in 1997. This also allowed the team to get experience on the eventual mechanisms deployed for maintenance and operations.

The initial NWChem development platform was a KSR-2 – this system was appropriate for exploration of programming models for both shared and distributed memory implementations. The first actual production hardware (an IBM SP2) and, in fact, all subsequent production hardware systems were purchased based on NWChem requirements and benchmarks. Since the initial development of NWChem, the program has been ported to a broad range of computer systems, from IBM systems running AIX, SGI IRIX and Altix systems, and HP systems running HPUX, Tru64 and Linux to Apple personal computers running OS X. In addition, the performance and capabilities of NWChem have increased substantially since version 1.0 was released in 1997. The latest version (4.7) includes many improvements to the algorithms used in NWChem as well as the ability to perform many new types of calculations.

Unlike many other supercomputing facilities, EMSL's Molecular Science Computing Facility supports a software development effort for NWChem and related software as well as supercomputer operations. The High Performance Software Development Group is presently led by Theresa Windus. Within this group, the Molecular Science Software project is responsible for the evolution, distribution, and support of Ecce, an extensible computational chemistry environment, and ParSoft, a set of software tools for massively parallel computers, as well as NWChem. In the High Performance Software Development Group, there are five computational chemists associated with evolution, distribution and support of NWChem—the same number (but not necessarily the same people) that were originally involved in the development of the software. It should be noted that the integration of NWChem with Ecce (the user interface) was much more difficult to achieve than originally anticipated. The integration process should probably have been initiated in 1995 (two years prior to the first official release in 1997).

## 5. Conclusion

Petascale computing is now a realizable goal that will impact all of science and engineering, not just those applications requiring the highest capability. But the optimum pathway to petascale science and engineering—the pathway that will realize the *full* potential of petascale computers to drive science and engineering—is unclear. Future computers cannot rely on continuing increases in clock speed to drive performance increases—heat dissipation problems will limit these increases. Instead, tomorrow's computing systems will include processors with multiple "processor cores" on each chip, special application accelerators, and reprogrammable logic devices (FPGAs). In addition, all of these types of processors may be included in a single system, interconnected by a high-performance communications fabric. Individual processors may even have heterogeneous "processor cores" in the fashion of the new Cell processor from IBM, Sony and Toshiba.[14] These technologies have the potential to dramatically increase the fidelity and range of computational simulations as well as the scope and responsiveness of data mining, analysis, and visualization applications. However, they also pose significant technical problems that must be addressed before their full potential can be realized.

So, the advances promised by petascale computers will not come gratis. The problems encountered in developing scientific codes for supercomputers with a performance exceeding 100-teraflops are technically complex, and their resolution will (once again) require an in-depth understanding of both the scientific algorithms and the computer hardware and systems software. Hardware problems to be overcome range from the memory bandwidth limitations of multicore microprocessor-based compute nodes to the utilization of "exotic" computing technologies (*e.g.*, FPGAs) to the bandwidth and latency limitations of the interprocessor communications fabric. Software problems to be overcome range from the choice of programming model to the development of numerical algorithms that scale t (at least!) tens of thousands of processors. And, in the end, we want a code that is extensible, portable, and maintainable. As the NWChem project illustrated, scientific codes that achieve these goals can be met by teams that include all of the needed expertise and that draw on talent both near and far. The pacing item for *petascale science and engineering*, as opposed to *petascale computing*, will be the state of the art in scientific applications.

As daunting as the above problems seem, it will be worth it! Combining the computing advances described above with advances in mathematical models and computational algorithms will lead to revolutionary new modeling and simulation capabilities. Problems that currently seem intractable will not only become doable, they will become routine. In chemistry, computational studies will become an integral and irreplaceable part of studies aimed at understanding the chemical processes involved in the environment, the burning of hydrocarbon fuels, and the industrial production of chemicals. The fidelity of modeling complex biomolecules will also take a major step forward, greatly increasing the contributions of computational chemistry to the life sciences. To realize these opportunities, however, the federal agencies must make investments in scientific simulation software, computing system software, and mathematical libraries necessary to capitalize on the potential of petascale computing.

[14] http://ww-03.ibm.com/chips/power/splash/cell/

# Supporting National User Communities at NERSC and NCAR

Timothy L. Killeen
National Center for Atmospheric Research

Horst D. Simon
NERSC Center Division, Ernest Orlando Lawrence Berkeley National Laboratory, University of California

## 1. Introduction

The National Energy Research Scientific Computing Center (NERSC) and the National Center for Atmospheric Research (NCAR) are two computing centers that have traditionally supported large national user communities. Both centers have developed responsive approaches to support these communities and their changing needs by providing end-to-end computing solutions. In this report we provide a short overview of the strategies used at our centers in supporting our scientific users, with an emphasis on some examples of effective programs and future needs.

## 2. Science-Driven Computing at NERSC

### 2.1 NERSC's Mission

The mission of NERSC is to accelerate the pace of scientific discovery by providing high performance computing, information, data, and communications services for research sponsored by the DOE Office of Science (DOE-SC). NERSC is the principal provider of high performance computing services for the capability needs of Office of Science programs — Fusion Energy Sciences, High Energy Physics, Nuclear Physics, Basic Energy Sciences, Biological and Environmental Research, and Advanced Scientific Computing Research.

Computing is a tool as vital as experimentation and theory in solving the scientific challenges of the 21st century. Fundamental to the mission of NERSC is enabling computational science of scale, in which large, interdisciplinary teams of scientists attack fundamental problems in science and engineering that require massive calculations and have broad scientific and economic impacts. Examples of these problems include global climate modeling, combustion modeling, magnetic fusion, astrophysics, computational biology, and many more. NERSC uses the Greenbook process[1] to collect user requirements and drive its future development.

[1] Simon, H.D. et al. "Science Driven Computing: NERSC's Plan 2006 – 2010," LBNL Report 57582, Berkeley, California, May 2005

Lawrence Berkeley National Laboratory (Berkeley Lab) operates and has stewardship responsibility for NERSC, which, as a national resource, serves about 2,400 scientists annually throughout the United States. These researchers work at DOE laboratories, other Federal agencies, and universities (over 50% of the users are from universities). Computational science conducted at NERSC covers the entire range of scientific disciplines but is focused on research that supports DOE's missions and scientific goals.

### 2.2 A Science-Driven Strategy to Increase Scientific Productivity

Since its founding in 1974, NERSC has provided systems and services that maximize the scientific productivity of its user community. NERSC takes pride in its reputation for the expertise of its employees and the high quality of services delivered to its users. To maintain its effectiveness, NERSC proactively addresses new challenges. We observe three trends that NERSC needs to address over the next several years:

- the widening gap between application performance and peak performance of high-end computing systems
- the recent emergence of large, multidisciplinary computational science teams in the DOE research community
- the flood of scientific data from both simulations and experiments, and the convergence of computational simulation with experimental data collection and analysis in complex workflows.

NERSC's responses to these trends are the three components of the science-driven strategy that NERSC will implement and realize in the next five years; ***science-driven systems, science-driven services,*** and ***science-driven analytics*** (Fig. 1). This balanced set of objectives will be critical for the future of the enterprise and its ability to serve the DOE scientific community.

- ***Science-Driven Systems:*** Balanced introduction of the best new technology for complete computational systems — computing, storage, networking, visualization and analysis.
- ***Science-Driven Services:*** The entire range of support activities, from high-quality operations and user services to direct scientific support, that enable a broad range of scientists to effectively use NERSC systems in their research. NERSC will concentrate on resources needed to realize the promise of the new, highly scalable architectures for scientific discovery in multidisciplinary computational science projects.
- ***Science-Driven Analytics:*** The architectural and systems enhancements and services required to integrate NERSC's powerful computational and storage resources to provide scientists with new tools to effectively manipulate, visualize, and analyze enormous data sets derived from both simulation and experiment.


Figure 1. Conceptual diagram of NERSC's plan for 2006–2010.

## Science-Driven Systems

Applications scientists have been frustrated by a trend of stagnating application performance relative to dramatic increases in claimed peak performance of high performance computing systems. This trend has been widely attributed to the use of commodity components whose architectural

designs are unbalanced and inefficient for large-scale scientific computations. It was assumed that the ever-increasing gap between theoretical peak and sustained performance was unavoidable. However, results from the Earth Simulator in Japan clearly demonstrate that a close collaboration with a vendor to develop a science-driven solution can produce a system that achieves a significant fraction of peak performance for critical scientific applications.

Realizing that effective large-scale system performance cannot be achieved without a sustained focus on application-specific systems development, NERSC has begun a science-driven systems strategy. The goal of this effort is to influence the vendors' product roadmaps to improve system balance and to add key features that address the requirements of demanding capability applications at NERSC — ultimately leading to a sustained Pflop/s system for scientific discovery. This strategy involves extensive interactions between domain scientists, mathematicians, computer experts, as well as leading members of the vendors' research and product development teams.

NERSC must be prepared for disruptive changes in processor, interconnect, and software technologies. Obtaining high application performance will require the active involvement of NERSC in understanding, driving, and adopting these technologies. The move towards open-source software will require additional efforts in software integration at NERSC.

The goal of the science-driven systems strategy is to enable new scientific discoveries, and that requires a high level of sustained system performance on scientific applications. The NERSC approach takes into account both credibility and risk in evaluating systems and will strike a balance between innovation and performance on the one hand and reliability on the other. While the discussion often focuses on the high-end platforms, NERSC will continue to emphasize maintaining Center balance, that is, improving all the systems at NERSC — storage, networking, visualization and analysis — commensurately with improvements in the high-performance computing platforms.

### Science-Driven Services

The DOE computational science community, in all its disciplines, has been organizing itself into large multidisciplinary teams. This trend was driven by the DOE Scientific Discovery through Advanced Computing (SciDAC) initiative, but has reached beyond the SciDAC teams. This trend has been driven by necessity as well as opportunity. The transformation became most apparent after massively parallel computers came to dominate the high end of available computing resources.

Technology trends indicate that the gap between the peak performance of next-generation systems and performance that is easily attainable could increase even more. NERSC has been focused on working with computational scientists to close this gap and help them scale their applications efficiently to current platforms. NERSC has formulated a science-driven services strategy that will address the requirements of these large computational science teams even more so than in the past, while at the same time maintaining the high level of support for all of its users.

### Science-Driven Analytics

A major trend occurring in computational science is the flood of scientific data from both simulations and experiments, and the convergence of experimental data collection, computational simulation, visualization, and analysis in complex workflows. Deriving scientific understanding from massive datasets produced by major experimental facilities is a growing challenge.

In recent years, NERSC has seen a dramatic increase in the data arriving from DOE-funded research projects. This data is stored at NERSC because NERSC provides a reliable long-term storage environment that assures the availability and accessibility of data for the community. NERSC has

helped accelerate this development by deploying Grid technology on all of its systems and by enabling and tuning high performance, wide area network connections to major facilities, for example the Relativistic Heavy Ion Collider at Brookhaven National Laboratory.

Now, NERSC must invest resources to complete an environment that allows easier analysis and visualization of large datasets derived from both simulation and experiment. Our third new thrust in science-driven analytics will enable scientists to combine experiment, simulation, and analysis in a coordinated workflow. This thrust will include activities enhancing NERSC's data management infrastructure, expanding NERSC's visualization and analysis capabilities, enhancing NERSC's distributed computing infrastructure, and understanding the analytical needs of the user community.

### 2.3 A Key Resource for the DOE Office of Science

In "Facilities for the Future of Science: A Twenty Year Outlook," the Office of Science has identified the need for creating new and/or improving on the current computational capability as a critical aspect of realizing its advanced scientific computing research vision.[2] It identified the NERSC upgrade as a near-term priority to ensure that NERSC, DOE's premier scientific computing facility for unclassified mission-critical research, continues to provide high-performance computing resources to support the requirements of scientific discovery.

[2] U.S. Department of Energy, Office of Science. Facilities for the Future of Science: A Twenty-Year Outlook, Washington, DC, November 2003.

As a high-end facility that serves all the DOE-SC programs with capability and high-end capacity resources, NERSC is a key resource in DOE-SC's portfolio of computing facilities. NERSC has established a reputation for providing reliable and robust services along with unmatched support to its users. Because of investments such as SciDAC, and the important role that computation will play in Genomics:GTL (formerly Genomes to Life) and the Nanoscale Science Research Centers, demands for computational resources in DOE-SC will continue to grow at a rapid rate, and NERSC's growth must keep pace. NERSC supports a large number (200–300) of projects of medium to large scale, occasionally requiring a very high capability resource, that fall within the mission of the Office of Science. The scientific productivity enabled by NERSC is demonstrated by the 2,206 papers in refereed publications in 2003 and 2004 that were based at least in part on work done at NERSC.

In NERSC's experience, there is a continuum of scientific computing systems and facilities. There are a few research groups with experienced users and very high computational requirements who are in a good competitive position to use a Leadership Class Facility. There is a much larger number of PIs and projects with high-end requirements who are best served by NERSC's high-end systems and comprehensive services, both of which distinguish NERSC from leadership computing and midrange computing centers, such as institutional or departmental clusters. Capability users include both single principal-investigator teams and community science teams. NERSC's science-driven services are important for both types of high-end users.

NERSC supports large-scale teams working on advanced modeling and simulation "community codes" whose development is shared by entire scientific research communities. These codes employ new mathematical models and computational methods designed to better represent the complexity of physical process and to take full advantage of current computational systems. NERSC provides focused support for these teams.

NERSC also supports single-PI teams consisting of a lead researcher and his or her group of collaborators, postdocs, and students, usually concentrated at a single location. For this class of users, NERSC's science-driven service is important because they are usually less knowledgeable

about computational technologies and they lack the resources to establish in-depth collaborations with computer science or mathematics experts. Computing at NERSC not only produces important scientific insights but also gives these users and teams the opportunity to advance to the leadership computing level for their most challenging computations.

As a centralized facility properly staffed and managed, NERSC provides the best possible mechanism for technology transfer between the computational efforts of different research programs. Moreover, a concentration of computing resources provides a more flexible mechanism to address changing priorities. SC's priorities for its programs sometimes change quickly because it is a mission agency. A general-purpose facility like NERSC, with a staff prepared to support the broadest possible array of scientific disciplines, allows DOE to switch priorities and quickly apply its most powerful computing resources to new challenges.

NERSC's role as a general scientific computing facility requires it to provide resources that are of common utility to the programs of the Office of Science. However, NERSC must be responsive to the specific needs of each program. Specific support for different programs, tailored to their varying needs, has been a key to the success of the center. Examples range from the collaborative effort of NERSC staff in scaling INCITE applications to 2,048 and 4,096 processors, to the operation of the PDSF cluster for the high energy and nuclear physics communities. The breadth of NERSC's support is best expressed by Figures 2 and 3, which summarize NERSC usage by discipline and institution.

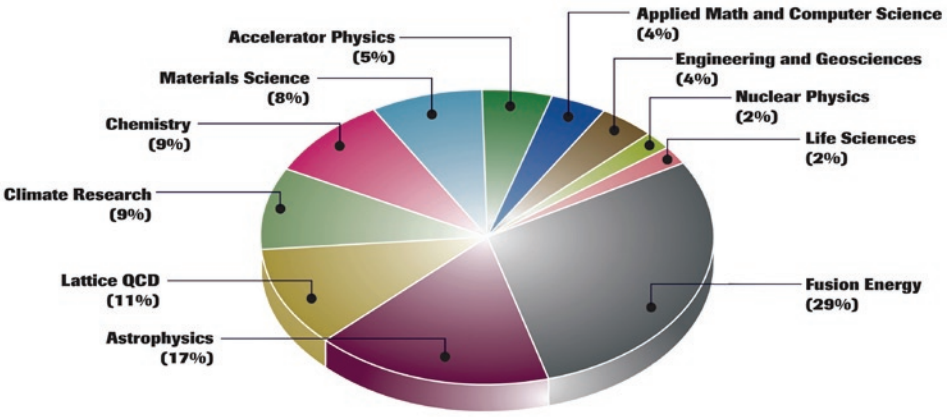


Figure 2. NERSC usage by scientific discipline for FY2004.

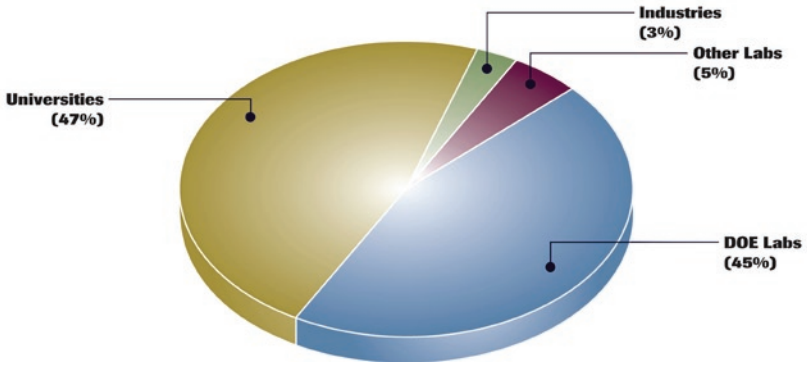


Figure 3. NERSC users by institution type for FY2004.

# 3. Science-Driven Computing at NCAR

## 3.1 NCAR's Mission

The mission of NCAR is to support, enhance, and extend the capabilities of the university community, nationally and internationally, to understand the behavior of the atmosphere and the global environment and to foster the transfer of knowledge and technology for the betterment of life on Earth.[3] NCAR is a principal provider of high performance computing services for the academic geosciences community in the United States and has a 48 year record of providing community supercomputing services. Over the years, NCAR and the community it serves has contributed centrally to

[3] NCAR as Integrator, Innovator, and Community Builder, the NCAR Strategic Plan, 2006-2016, http://www.ncar.ucar.edu/

- The scientific underpinnings of numerical weather forecast and climate modeling;
- The understanding of the coupled ocean/atmosphere climate system;
- The detailed chemistry of the stratosphere and troposphere;
- Solar magnetism, helioseismicity, and solar coronal mass ejections;
- The dynamics and chemistry of the upper atmospheres of earth and other planets;
- The microphysics of clouds and convective processes;
- The socioeconomic impacts of climate change and severe weather; and
- The role of human activities in causing and responding to large-scale Earth system change

Computing continues to be an essential part of NCAR's work and the center has a commitment to end-to-end services, spanning high-performance computing, application development and user support services, data management and data curation, visualization, networking, middleware, and all the components of what is commonly referred to as "cyberinfrastructure." The emphasis at NCAR is on solving computing problems related to the geosciences, and NCAR computational architecture acquisition and system support decisions are centered on the needs of this large but finite scientific domain. Human capital development is an essential part of this commitment.

In a similar fashion to NERSC, NCAR favors a balanced approach to high performance computing, stressing robust operational performance of diverse computing platforms with regular upgrade paths (Fig. 4), sophisticated application development, attention to software reuse and application portability with careful verification pathways, computational efficiency, redundant mass storage and secure data management systems. NCAR has experienced many of the same trends and challenges reported by NERSC, including the move to larger and more interdisciplinary teams of investigators, the need to close the gap between "sustained" and "peak" performance, and the requirement for matching the data system performance with application needs.

NCAR supports a "Community Model" approach that is perhaps unique among the large computational centers in the United States. This approach involves the development of well supported, open-source, large scope codes that have lifetimes of years to decades, are regularly enhanced and updated to reflect emerging scientific needs, and are managed and driven by the broad academic community, with NCAR playing the key coordinating role. NCAR's community models are freely available to all and are supported with help desks, version control systems, extensive documentation, regular user tutorials and workshops, and a significant body

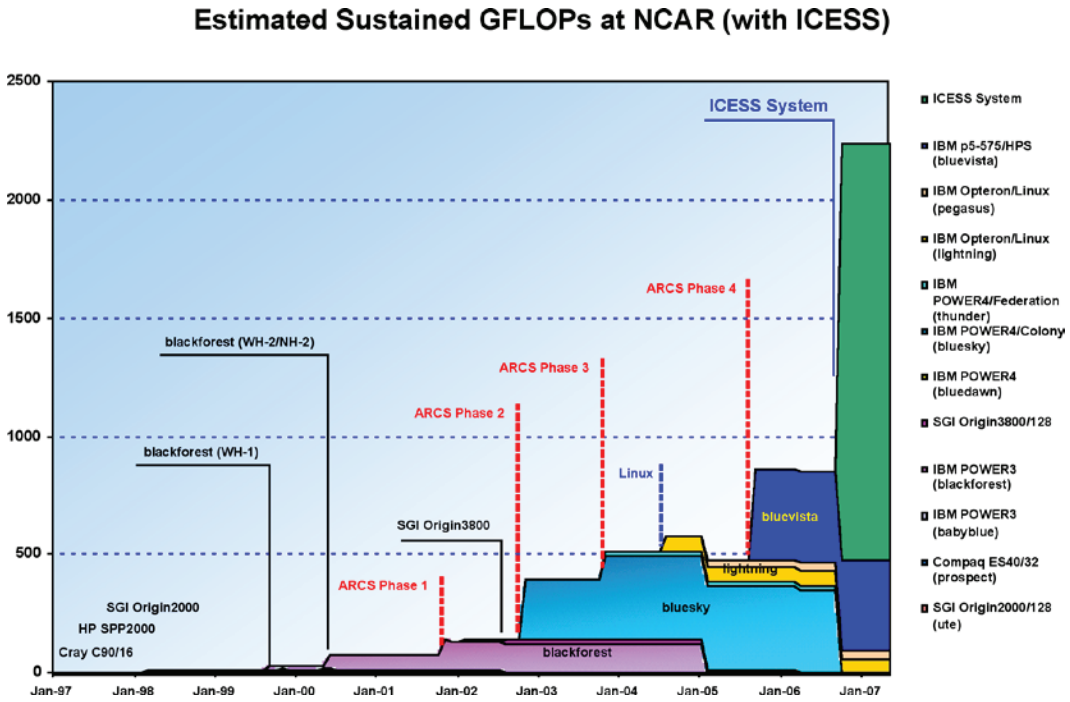


Figure 4. Sustained performance of applications running on NCAR computing platforms over the past 9 years. ICESS stands for the NCAR Integrated Computing Environment for Scientific Simulation, an ongoing procurement effort.

of peer-reviewed publications describing both computational and scientific aspects. Important examples of NCAR-managed community models include the Community Climate System Model, the Weather Research and Forecast Model, and the Earth System Modeling Framework. Brief descriptions of these three community science activities at NCAR are provided below to illustrate how NCAR supports national user communities.

### 3.2 The NCAR Community Climate System Model Program

The Community Climate System Model (CCSM)[4] is a comprehensive system for studying the past, present, and future of the Earth. In contrast to traditional weather-forecast models that focus only on the atmosphere, the CCSM includes components that simulate the evolution and interactions among the atmosphere, ocean, land surface, and sea ice. The principal objectives of the CCSM program are to develop a comprehensive numerical model with which to study the Earth's present climate, to investigate seasonal and inter-annual variability in the climate, to explore the history of the Earth's climate, and to simulate the future of the environment for policy formation.

[4] http://ww.ccsm.ucar.edu/

CCSM has been designed with input from a broad community of climate scientists, computer scientists, and software engineers. This community also shares the scientific code and results produced by the model. In fact, CCSM is the only climate model that is developed as open source code and is distributed via the web to the world-wide climate community. CCSM is funded with support from the National Science Foundation (NSF), DOE, the National Space and Aeronautics Administration (NASA), and the National Oceanographic and Atmospheric Administration (NOAA). The CCSM community includes some 900 members located at universities and laboratories throughout the world.

In order to support a broad community, CCSM must operate both as a research and an operational climate model, and therefore must be easily portable to a wide range of computa-

tional platforms. CCSM or its components can be run "out of the box" on a variety of Linux clusters, Apple servers, SGI Origin and Altix systems, and IBM and Intel clusters. It has also been enabled on NEC and Cray vector supercomputers, IBM Power-series clusters, and Cray clusters of scalar processors. The developers are now exploring modifications to CCSM to ensure efficient execution on other massively parallel architectures. The CCSM team has developed a comprehensive suite of tests to ensure that the model algorithms work reliably and transparently across such a heterogeneous computing environment.

CCSM is designed to be flexible and extensible, an important characteristic since it will serve as a basis for the development of a more complete Earth System model over the next several years. This Earth System model will simulate the chemical, biogeochemical, and physical state of the climate system. The CCSM development effort is managed by a Scientific Steering Committee with membership from the broad academic research community, as well as from NCAR.

The CCSM results for the IPCC provide a sobering look into the future of the planet and are being documented in more than 200 peer reviewed scientific publications. Figure 5 shows projections of the time evolution of summer Arctic ice area for several IPCC greenhouse gas forcing scenarios. Note that summer ice is projected to disappear from the Arctic toward the latter part of this century under the IPCC "A2" scenario for socio-economic development.

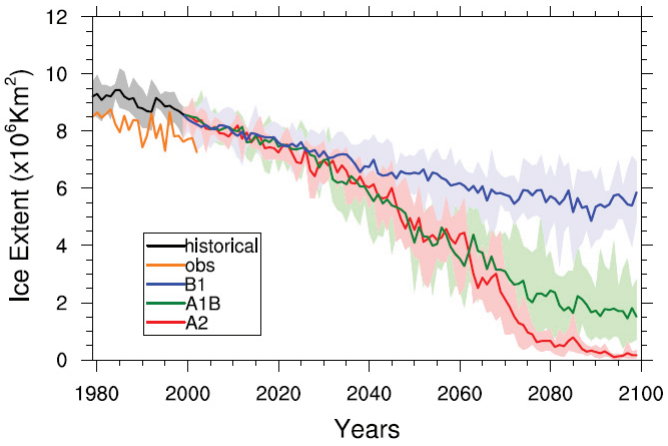


Figure 5. CCSM IPCC ensemble simulations of Arctic ice extent for the next century.[5] The individual curves represent IPCC scenarios and the shaded regions provide the uncertainty bounds from the multiple realizations.

[5] Teng, H, W.M. Washington, G. A. Meehl, L. Buja and G. Strand 2006: 21st Century Arctic Climate Change Simulated by CCSM3 IPCC Scenarios, Clim. Dyn., Doi: 10.1007/s00382-005-0099-z

### 3.3 The Weather Research and Forecast Model

A long-time focus in numerical modeling of the atmosphere has been the development and improvement of capabilities that can simulate the conditions that dictate the weather. Such systems are typically called "mesoscale" atmospheric models, where mesoscale refers to the spatial dimension over which most of the weather that influences daily, human activity occurs. NCAR has been developing a new numerical weather prediction (NWP) model that is now coming into its own: the Weather Research and Forecasting Model (WRF).[6] WRF is employed worldwide with the largest number of registered users (over 3,700) for any such model today.

[6] http://www.wrf-model.org/

The WRF model is different from existing NWP technologies in a number of ways. Rather than created by a single researcher, institution, or agency, WRF was developed in the U.S. through a partnership of both research and operational (i.e., official weather forecasting) groups. The initial development began in 1997, and the partners have been NCAR, the U.S.

National Centers for Environmental Prediction (NCEP), the U.S. Air Force Weather Agency, the U.S. Navy's Naval Research Laboratory, the NOAA's Earth System Research Laboratory, the Federal Aviation Administration (FAA), and Oklahoma University. The goal was to create an NWP tool for use by both the operational and research meteorological communities. A key motivation was having a vehicle that, with relative ease and rapidity, could make the latest in research advances available to public forecasting.

The WRF modeling system features a software framework that is modular, plugin-compatible, and allows portability to a wide range of computer architectures. It runs on hardware from laptops, to desktop workstations, to PC Linux clusters, to high-performance supercomputers. WRF is parallelized and is efficient in massively parallel, distributed-memory environments. The software framework permits ease of coupling with other earth system numerical models (e.g., ocean circulation codes or air chemistry modules). WRF also provides sophisticated data assimilation— the incorporation of observed meteorological information from satellites and other observing systems

WRF is currently being used for official forecasting in the U.S. by NCEP, which provides NWP model guidance for the forecasters of the National Weather Service. On the research side, WRF's applications range from study of atmospheric processes and weather from the tropics to the poles. Targets of special interest for WRF so far have been severe thunderstorms and powerfully damaging hurricanes, given their enormous societal impacts in the U.S. For the past three hurricane seasons, for example, WRF has been run at NCAR in real-time to offer high-resolution (i.e., detailed) forecasts of storms, which have threatened landfall. Figure 6 offers an example of how well WRF can depict one of these monsters. Successes such as this are demonstrating that WRF is fulfilling its promise as the pre-eminent next-generation numerical weather prediction model.

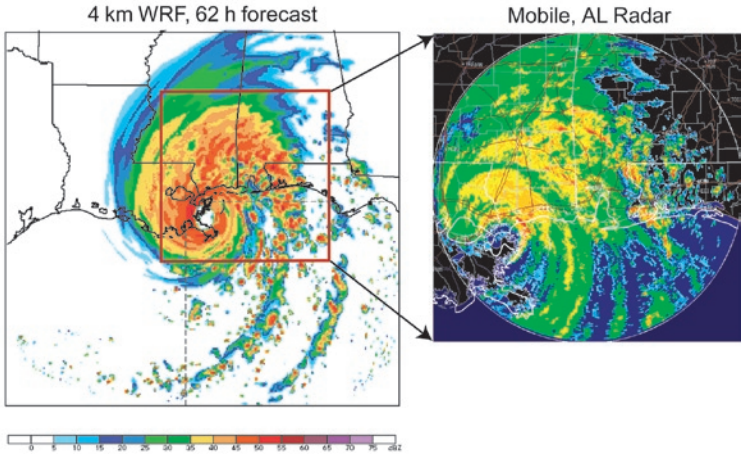


Figure 6. WRF simulation of Hurricane Katrina computed 3-days before landfall (left), compared with later radar observations of the actual landfall (right).

### 3.4 The Earth System Modeling Framework

In another example of NCAR-supported community systems, The Earth System Modeling Framework (ESMF)[7] provides a high performance common modeling infrastructure for climate and weather models and is widely available as a community-owned and managed product. It is in active use by groups working with hydrology, air quality, and space weather models. ESMF

[7] http://ww.esmf.ucar.edu/

is the technical foundation for the NASA Modeling, Analysis, and Prediction (MAP) Climate Variability and Change program and the DoD Battlespace Environments Institute (BEI). It has been incorporated into the CCSM, the WRF model, and many other applications.

The key concept that underlies both ESMF is that of *software components*. Components are software units that are "composable," meaning they can be combined to form coupled applications. These components may be representations of physical domains, such as atmospheres or oceans; processes within particular domains such as atmospheric radiation or chemistry; or computational functions, such as data assimilation or I/O. ESMF provides interfaces, an architecture, and tools for structuring components hierarchically to form complex, coupled modeling applications. ESMF components may be run sequentially, concurrently, or in a mixed mode on computers ranging from laptops to the world's largest supercomputers. The ESMF project encourages a new paradigm for geosciences modeling: one in which the community can draw from a federation of many interoperable components in order to create and deploy modeling applications. The goal is to enable a rich network of collaborations and a new generation of models that can simulate the Earth's environment and predict its behavior better than ever before.

ESMF is an open source project that is actively reaching out to universities, national laboratories, industry, and the international community. ESMF is funded by a collection of agencies, and its development priorities and direction are set by multi-agency management bodies. Although the core development team is located at NCAR, the ESMF code has a growing number of contributors from collaborating sites. The project has been remarkably successful in its ability to bring disparate groups together, from the developer level all the way up to the agency level, and to get them working towards the common goal of better models.

Because of the success of the CCSM, WRF and ESMF and other similar community projects, NCAR is considering an overarching effort to develop an "Earth System Knowledge Environment." This environment would combine the key functions of all these programs and would lead to a fully supported and integrated "workspace" for modeling, computation, analysis, data management, data assimilation, and end-user diagnostics for the international community of geoscientists and societal decision makers charged with understanding the Earth System and its variability.

## 4. Summary

A strong emphasis on community involvement and governance has been critical to the success of NERSC and NCAR and is also central to plans for the future for both centers. NERSC and NCAR both support broad communities that are poised to make major breakthroughs in knowledge and understanding in very important scientific fields. Careful optimization of resources and capabilities will undoubtedly require continued attention and creativity as new computational systems develop and propagate. Both centers are ready to meet the challenge.

**Acknowledgements**
This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research of the U.S. Department of Energy under Contract No. DE-AC 03-76SF00098. NCAR is operated by the University Corporation for Atmospheric Research under sponsorship of the National Science Foundation.

One of the authors (Killeen) acknowledges important assistance from Al Kellie, Jordan Powers, Cecelia DeLuca, Marika Holland, Bill Collins, Jim Hack, and Veda Emmett in the development of this report.

CTWatch QUARTERLY

ISSN 1555-9874 Volume 2 Number 2 **May 2006**

# Designing and Supporting Science-Driven Infrastructure

GUEST EDITORS: **FRAN BERMAN** AND **THOM DUNNING**

AVAILABLE ON-LINE:
www.ctwatch.org/quarterly/

E-MAIL CTWatch QUARTERLY:
quarterly@ctwatch.org

**CTWATCH IS A COLLABORATIVE EFFORT**

http://icl.cs.utk.edu/

http://www.ncsa.uiuc.edu/

http://www.sdsc.edu/

**CTWATCH IS A PUBLICATION OF THE CYBERINFRASTRUCTURE PARTNERSHIP**

www.ci-partnership.org

**SPONSORED BY**