Available online at www.sciencedirect.com
SCIENCE DIRECT®
Parallel Computing 30 (2004) 1001–1017
PARALLEL COMPUTING
www.elsevier.com/locate/parco

Molecular structure determination on a computational and data grid

Mark L. Green, Russ Miller *

Department of Computer Science and Engineering, Center for Computational Research, State University of New York, Buffalo, NY 14260, USA

Received 20 April 2004; revised 1 July 2004; accepted 15 July 2004

Abstract

The focus of this paper is on the design and implementation of a critical program in structural biology onto two computational and data grids. The first is the Buffalo-based ACDC grid, which uses facilities at several research institutions in Western New York. The second is Grid2003, the iVDGL grid established late in 2003 primarily for physics and astronomy applications. In this paper, we present an overview of the ACDC Grid and Grid2003, focusing on the implementation of several new ACDC computational and grid tools.

Keywords: Structure determination; Computational Grids; Data Grids; ACDC Grid; Grid2003; Shake-and-Bake; Molecular structure

1. Introduction

The ACDC-Grid [11,20,21] is a proof-of-concept general-purpose grid implemented in Buffalo, NY. In this paper, we focus on the ability of the ACDC-Grid to enable a cost-effective and ubiquitous solution to a critical application in structural biology that is used to determine molecular structures from X-ray

* Corresponding author.
E-mail addresses: mlgreen@ccr.buffalo.edu (M.L. Green), miller@buffalo.edu (R. Miller).

0167-8191/$ - see front matter
doi:10.1016/j.parco.2004.07.008

crystallographic data. *SnB* [1], a computer program based on the *Shake-and-Bake* method [2,3], is the program of choice for structure determination in many of the 500 laboratories that have acquired it [4–6]. This computationally-intensive procedure can exploit the grid's ability to present the user with a large computational infrastructure that will allow for the processing of a large number of related molecular trial structures [7,8].

The *SnB* program uses a dual-space direct-methods procedure for determining crystal structures from X-ray diffraction data. This program has been used in a routine fashion to solve difficult atomic resolution structures, containing as many as 1000 unique non-hydrogen atoms, which could not be solved by traditional reciprocal-space routines. Recently, the focus of the *Shake-and-Bake* research team has been on the application of *SnB* to solve heavy-atom and anomalous-scattering substructures of much larger proteins provided that 3–4 Å diffraction data can be measured. In fact, while traditional direct methods had been successfully applied to substructures containing on the order of a dozen selenium sites, *SnB* has been used to determine as many as 180 selenium sites. Such solutions have led to the determination of complete structures containing hundreds of thousands of atoms.

The *Shake-and-Bake* procedure consists of generating structure invariants and coordinates for random-atom trial structures. Each such trial structure is subjected to a cyclical automated procedure that includes computing a Fourier transform to determine phase values from a proposed set of atoms (initially random), determining a figure-of-merit [9], refining phases to locally optimize the figure-of-merit, computing a Fourier transform to produce an electron density map, and employing a peak-picking routine to examine the map and find the maxima. These peaks are then considered to be atoms, and the cyclical process is repeated for a predetermined (by the user) number of cycles.

The running time of this procedure ranges from minutes on PCs to months on supercomputers. Trial structures are continually and simultaneously processed, with the final figure-of-merit values of each structure stored in a file. The user can review a dynamic histogram during the processing of the trials in order to determine whether or not a solution is likely present in the set of completed trial structures.

2. Grid overview

The Computational Grid represents a rapidly emerging and expanding technology that allows geographically distributed resources (CPU cycles, data storage, sensors, visualization devices, and a wide variety of Internet-ready instruments), which are under distinct control, to be linked together in a transparent fashion [10,22,23]. The power of the Grid lies not only in the aggregate computing power, data storage, and network bandwidth that can readily be brought to bear on a particular problem, but on its ease of use. Since resources in a grid are pooled from many different domains, each with its own security protocol, insuring the security of each system on the Grid is of paramount importance.

Grids are now a viable solution to certain computationally and data-intensive computing problems for the following reasons: (a) The Internet is reasonably mature and able to serve as fundamental infrastructure. (b) Network bandwidth has increased to the point of being able to provide efficient and reliable services. (c) Storage capacity has now reached commodity levels, where one can purchase a terabyte of disk for roughly the same price as a high-end PC. (d) Many instruments are Internet-aware. (e) Clusters, supercomputers, storage and visualization devices are becoming more easily accessible. (f) Applications have been parallelized. (g) Collaborative environments are moving out of the alpha phase of implementation.

For these and other reasons, grids are starting to move out of the research laboratory and into early-adopter production systems. The focus of grid deployment continues to be on the difficult issue of developing high quality middleware.

Many types of computational tasks are naturally suited to grid environments, including data-intensive applications. Research and development activities relating to the Grid have generally focused on applications based on data that is stored in files. However, in many scientific and commercial domains, database management systems have a central role in data storage, access, organization, and authorization for numerous applications. Part of our research effort is targeted at enabling systems that are more accessible within a grid framework.

As Grid computing initiatives move forward, issues of interoperability, security, performance, management, and privacy need to be carefully considered. In fact, security is especially critical in terms of authentication and authorization in Grid environments so as to insure application and data integrity. Best practice scheduling and resource management documents, protocols, and API specifications to enable interoperability are currently being developed by a variety of organizations. Note that several layers of security, data encryption, and certificate authorities already exist in grid-enabling toolkits such as Globus Toolkit 3 [24].

3. Advanced Computational Data Center (ACDC) grid development

The development of the Advanced Computational Data Center Grid (ACDC-Grid) portal focuses on establishing an extensible and robust Application Programming Interface (API) that uses Grid-enabling Application Templates (GATs) [11]. The ACDC-Grid GATs define standard procedures that many scientific applications require when executing in a grid environment. There are several grid metrics and components that are required for defining a GAT that will be presented in this paper. Fig. 1 shows the ADCD web portal, which is the single point of contact for all ACDC-Grid computational and data grid resources.

The ACDC-Grid is based on the Globus Toolkit middleware version 2.2.4 and the web portal is served by Apache HTTP Server version 2.0. All of the web portal pages are dynamically created using the PHP hypertext preprocessor scripting language, JavaScript, and real-time MySQL database access. Each web portal page also allows the strict use of security and authentication procedures, defining a fine grained custom interface for each grid user. Several grid user Access Control Levels (ACLs)

Fig. 1. The ACDC-Grid web portal user interface.

have been defined for un-authenticated, general, system administrator, and grid administrator web portal page access.

The base Globus Toolkit middleware Metacomputing Directory Service (MDS) information is stored in the ACDC-Grid database and can be queried directly or displayed in a Java tree from the web portal, as shown in Fig. 2.

The ACDC-Grid job monitoring system is designed to be an extremely lightweight and non-intrusive tool for monitoring applications and resources on computational grids. It also provides a historical retrospective of the utilization of such resources, which can be used to track efficiency, adjust grid-based scheduling, and perform a predictive assignment of applications to resources.

The ACDC-Grid database aggregates compute platform statistics on an arbitrary time scale (e.g., 5, 15, 30, 60 min), including load, running/queued jobs, backfill availability, queue schedule, and compute platform production rate. This information can

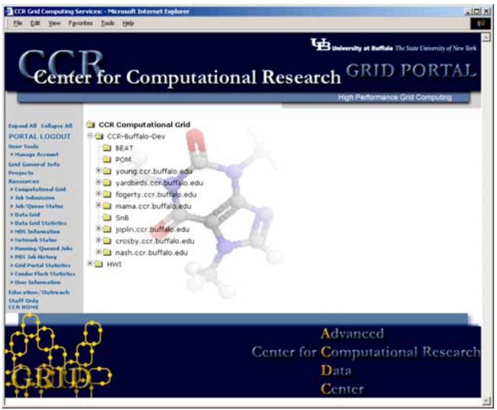

Fig. 2. The ACDC-Grid Metacomputing Directory Service web portal with Java tree view.

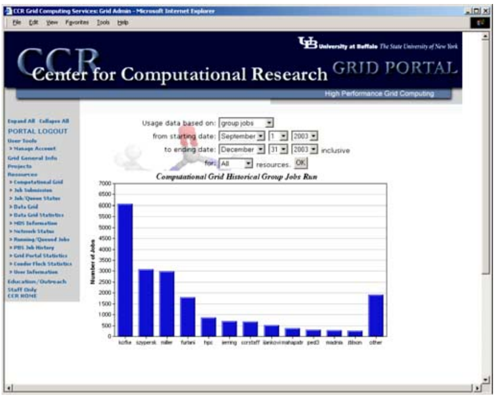

Fig. 3. ACDC-Grid historical chart of compute platform jobs completed on all platforms during the period of 1 September through 31 December 2003.

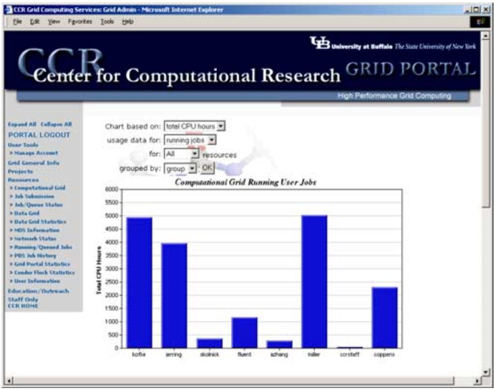

Fig. 4. ACDC-Grid running/queued jobs CPU consumption based on user groups for all resources.

be queried directly by the user and presented in chart form for both historical grid jobs, as shown in Fig. 3, and running or queued jobs, as shown in Fig. 4.

The intelligent management of consumable resources, including compute cycles, requires accurate up-to-date information. The ACDC job monitoring system provides near real-time snap shots of critical computational job metrics, which are stored in a database and presented to the user via dynamic web pages. Jobs are segmented into several classes (e.g., running jobs, queued jobs, and so forth) and statistics for each class are created on the fly.

As shown in Fig. 5, a variety of critical information is available to the user, including the total number of currently running jobs, the total CPU hours consumed by these jobs, the average number of nodes per job, the average runtime per job, and so forth. Additional information presented includes raw values and the percentage attributed to the user or group of concern. (As shown in the example presented in

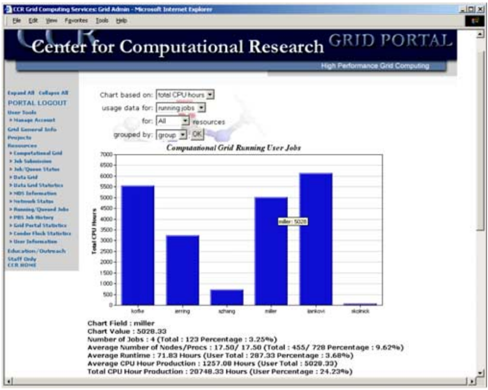

Fig. 5. ACDC-Grid metrics and statistics of jobs currently running on all compute platforms.

Fig. 5, information is presented for the aggregate of all uses in the *miller* group.) It should be noted that from the pull-down menus the user has many options in terms of the data to be presented. That is, the chart can be based on total jobs, total CPU hours, or total runtime. The chart can include information based on running or queued jobs. The resources considered can be a complete grid or a subset of a grid. Finally, the resources can be compiled with respect to a user, group, virtual organization, or queue.

Furthermore, all metrics obtained from a given resource are time stamped so that the age of the information is available. In order to present voluminous amounts of data in a hierarchical fashion, all top-level ACDC Job Monitoring charts have a "drill down" feature that gives the user increasingly more detailed information. This feature is essential when a post-mortem analysis is utilized in order to assess the performance of the Grid (e.g., the number of CPU hours consumed over a given period of time across all available resources). The ability to drill down for additional information is also valuable when querying information about running or queued jobs, as shown in Fig. 6.

The ACDC-Grid (Fig. 7) is a heterogeneous collection of compute platforms using several different native queue managers (e.g., OpenPBS, Condor, fork), a variety of operating systems (e.g., RedHat Linux, IRIX, Windows) and a variety of Wide Area Network (WAN) connections (e.g., GigE, Fast Ethernet, T1).

The ACDC computational and data grid performance often depends on the availability, latency, and bandwidth of the WAN. Thus, all compute platforms use the Network Weather Service (NWS) [25] for reporting the essential latency and bandwidth statistics to the database. This information can be presented to the user, as shown in Fig. 8. This information is also directly available to the ACDC-Grid GATs in order to efficiently manage the computational and data grids.

In an increasing number of scientific disciplines, large data collections are emerging as an important community resource. The ACDC data grid complements the ACDC computational grid in terms of managing and manipulating these data

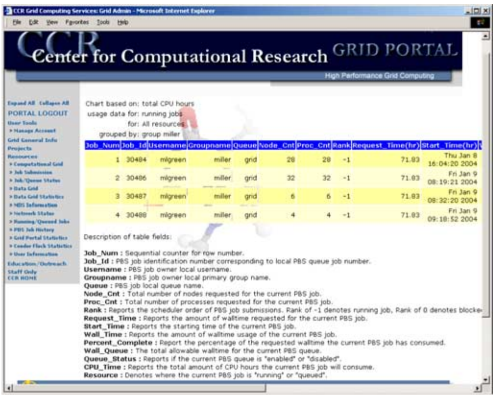

Fig. 6. ACDC-Grid additional job information dynamically queried from the database.

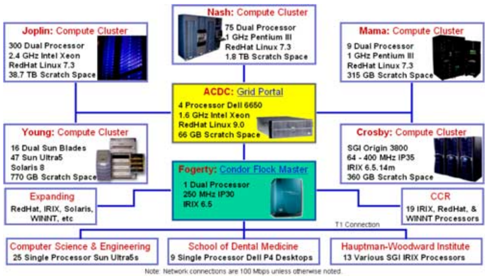

Fig. 7. ACDC-Grid computational resources.

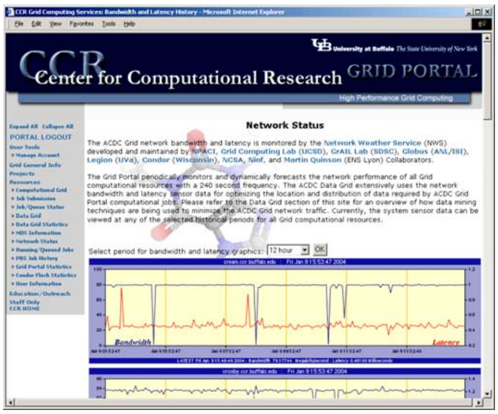

Fig. 8. ACDC-Grid network status can be presented for any given period of time for all compute platforms.

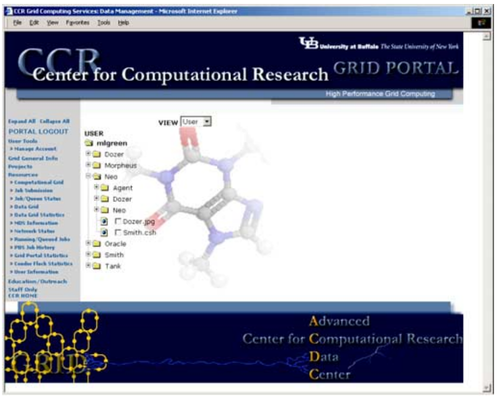

Fig. 9. ACDC data grid Java tree view of files.

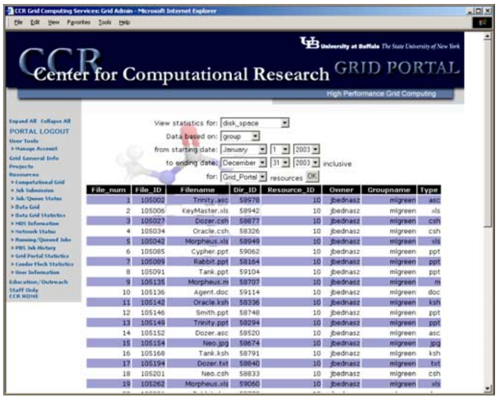

Fig. 10. ACDC data grid list-based view of sorted user files.

collections. A *data grid* denotes a large network of distributed storage resources, such as archival systems, caches, and databases, which are logically linked so as to create a sense of global persistence. The goal of the ACDC data grid is to transparently manage data distributed across heterogeneous resources, providing access via a uniform (web) interface, as shown in Fig. 9. In addition, we would also like to enable the transparent migration of data between various resources while preserving uniform access for the user. Maintaining metadata information about each file and its location in a global database table is essential.

The hierarchical display does not list the file attribute data, so a list-based display has also been developed that can be used for sorting data grid files based on available metadata (e.g., filename, file size, modification time, owner, etc.), as shown in Fig. 10.

Basic file management functions are available via a platform-independent user-friendly web interface that includes file transfer capabilities, a simple web-based file editor, an efficient search utility, and the logical display of files for a given user in three divisions (user/group/public). Collating and displaying statistical information is particularly useful to administrators for optimizing usage of resources. The ACDC data grid infrastructure periodically migrates files between data repositories for optimal usage of resources. The file migration algorithm depends on a number of factors, including

- user access time,
- network capacity at time of migration,
- user profile, and
- user disk quotas on various resources.

Further, we have the ability to mine log files, which aids in the determination of

- the amount of data to migrate in one cycle,
- the appropriate migration cycle length,
- the file access pattern of a data grid user, and
- the access pattern for public or group files.

The user global file-aging attribute is indicative of a user's access across their own files and is an attribute of a user's profile. The local file aging attribute is indicative of overall access of a particular file by users having group or public access. The latter is an attribute of a file and is stored in the file management data grid table. During migration, these attributes are used to determine the files that are to be migrated from the grid portal repository to a remote resource repository. Specifically, file migration is a function of global file aging, local file aging, and resource usage (e.g., the previous use of files on individual compute platforms is a factor in determining file migration). By tracking the file access patterns of all user files and storing this information in the associated database tables, the ACDC data grid infrastructure can automatically determine an effective repository distribution of the data grid files. See Fig. 11 for a schematic of the physical data ACDC data grid.

Support for multiple access to files in the data grid has been implemented with file locking and synchronization primitives. The ACDC data grid also provides security for authentication and authorization of users, as well as policies and facilities for

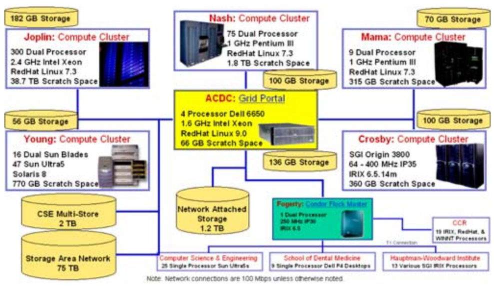

Fig. 11. ACDC data grid repository location, network bandwidth, and size.

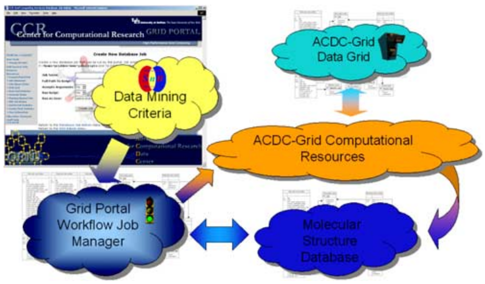

Fig. 12. ACDC-Grid grid-enabled data mining diagram.

data access and publication. The ACDC data grid algorithms are continually evolving to minimize network traffic and maximize disk space utilization on a per user basis. This is accomplished by data mining user usage and disk space requirements in a ubiquitous and automated fashion.

The *SnB* grid-enabled data mining application utilizes most of the ACDC-Grid infrastructure that we have presented. A typical *SnB* job uses the Grid Portal to supply the molecular structures parameter sets to optimize, the data file metadata, the grid-enabled *SnB* mode of operation (dedicated or back fill), and the *SnB* termination criteria. The Grid Portal then assembles the required *SnB* application data and supporting files, execution scripts, database tables, and submits jobs for parameter optimization based on the current database statistics. ACDC-Grid job management automatically determines the appropriate execution times, number of trials, number of processors for each available resource, as well as logging the status of all concurrently executing resource jobs. In addition, it automatically incorporates the *SnB* trial results into the molecular structure database, and initiates post-processing of the updated database for subsequent job submissions. Fig. 12 shows the logical relationship for the *SnB* grid-enabled data mining routine described.

4. ACDC-grid web portal user and resource administration

The infrastructure for the ACDC-Grid Web Portal is based on a central data repository and web scripting language. This infrastructure provides dynamic control of the web portal, as well as generating dynamic portal content based on real-time information and access control, as shown in Fig. 13.

Fig. 14 shows how an authenticated grid user can access a simple form for updating current information.

The grid administrators have many tools available for managing individual or multiple users at any given time. As shown in Fig. 15, grid user accounts can be selected by using a fuzzy search on "last name" or "organization", or by the "date added" or "date of last login". The process of creating user accounts is highly automated, requiring human input only in deciding whether or not a request should be accepted or rejected and what initial permissions to grant the user.

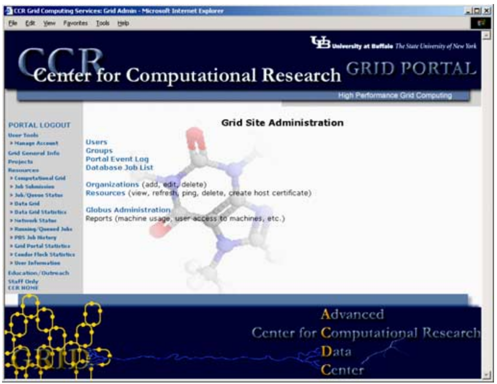

Fig. 13. ACDC-Grid portal grid site administration.

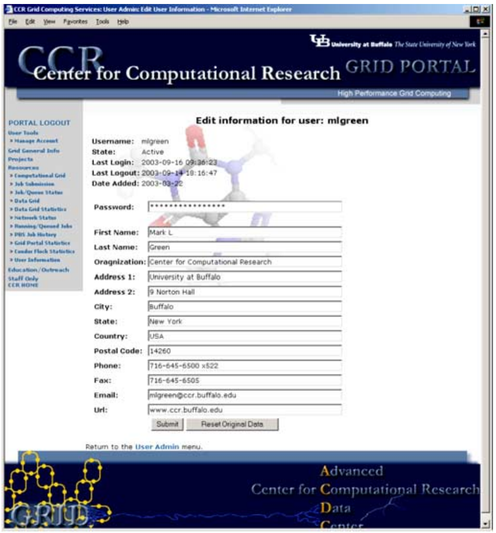

Fig. 14. ACDC-Grid portal grid user information form.

The entire grid user list can also be sorted by information entered through the user information form. Granting a grid user permission to use grid-enabled computational sources, programs, and data repositories is managed through a point and click Java dynamic interface, as shown in Fig. 16.

The administrator can grant permission to a user for access to available resources. The grid infrastructure also allows administrators to monitor the state of the resources in the grid, user job submission, and the state of the portal itself. Please refer to Fig. 17.

Using the data repository, the infrastructure provides a tool that can be used to notify administrators of possible issues that arise in the grid. The computational

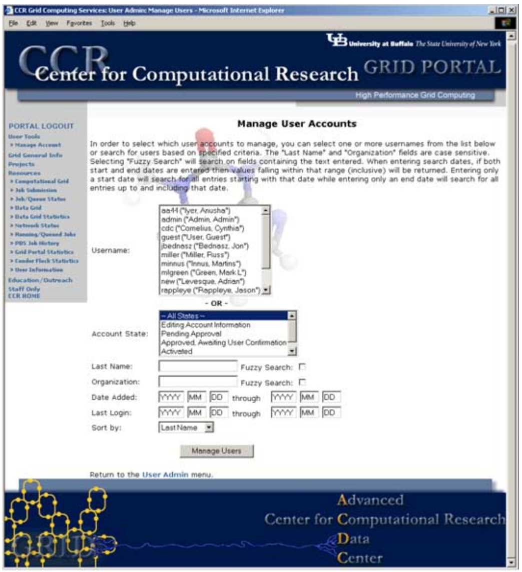

Fig. 15. ACDC-Grid portal manage user accounts form.

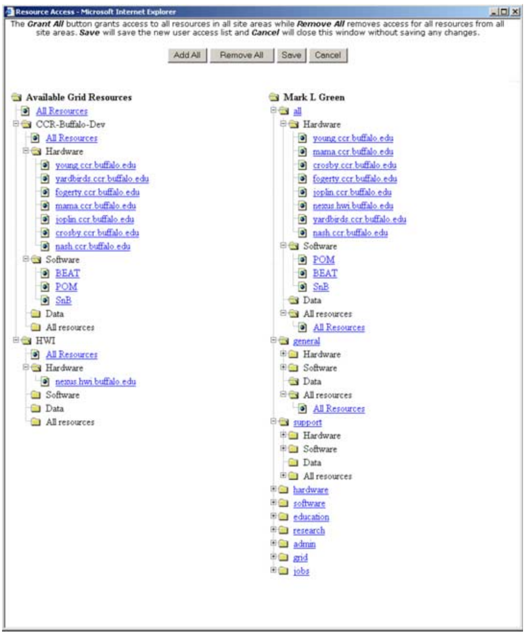

Fig. 16. ACDC-Grid portal Java dynamic interface.

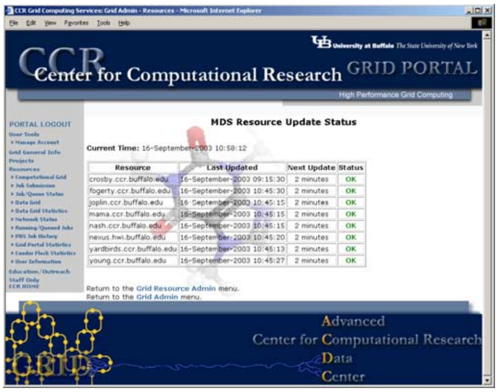

Fig. 17. ACDC-Grid portal manage user accounts form.

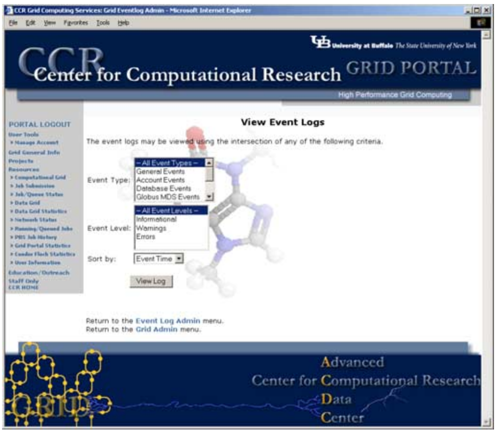

Fig. 18. ACDC-Grid portal event log management form.

and data grid administration is assisted through the use of automated portal job scripts that are stored in the database and executed based on pre-defined conditions in the grid. As shown in Fig. 18, a comprehensive event log may be viewed and mined for information or error conditions.

Trends in user activity and resource utilization may be monitored and mined over days, months, and years for computational and data grid resources. Finally, the portal infrastructure uses templates and the data repository to create pages dynamically based on a user's access rights, the current state of the resources in the grid, the organization that the user belongs to, and the user's previous activity. The portal infrastructure provides users with many features including the ability to define job workflows, save workflows as templates to be re-used at a later date, examine both current and historical job information, and view the load and availability of the grid resources.

5. Grid2003 participation experience

The International Virtual Data Grid Laboratory (iVDGL) is a global Data Grid that provides resources for experiments in physics and astronomy [12]. Its computing, storage, and networking resources in the US, Europe, Asia, and South America provide a unique computational laboratory that has been assembled to test and validate Grid technologies at international and global scales. The Grid2003 project [13] was defined and planned by Stakeholder representatives in an effort to align iVDGL project goals with the computational projects associated with the Large Hadron Collider (LHC) experiments.

The Grid Laboratory Uniform Environment (GLUE) [14] collaboration was created in February 2002 to provide a focused effort to achieve interoperability between the US physics Grid projects and the European projects. Participant US projects include iVDGL, Grid Physics Network (GriPhyN) [26], and Particle Physics Data Grid (PPDG) [15]. Participant European projects include the European Data Grid (EDG) Project [16], Data TransAtalantic Grid (DataTAG) [17], and CrossGrid [18]. Since the initial proposal for the GLUE project, the LHC Computing Grid (LCG) project was created at CERN [19] to coordinate the computing and Grid software requirements for the four LHC experiments, with a goal of developing common solutions. One of the main project goals is deploying and supporting global production Grids for the LHC experiments, which resulted in the Grid2003 "production" grid. See Fig. 19.

5.1. Goals of the Grid2003 project

The iVDGL Steering Committee set the following broad goals for the Grid2003 project:

Fig. 19. Grid2003 project web page.

- Provide the next phase of the iVDGL Laboratory.
- Provide the infrastructure and services needed to demonstrate LHC production and analysis applications running at scale in a common grid environment.
- Provide a platform for computer science technology demonstrators.

The ACDC-Grid, *SnB*, and several of the tools introduced in this paper, have been incorporated into Grid2003. The goals of Grid2003 included meeting a set of performance targets, using metrics listed in a planning document. The central project milestone can be summarized as delivery of a shared, multi-Virtual Organization (VO), multi-application, grid laboratory in which performance targets were pursued through deployment and execution of application demonstrations during the period before, during, and after the SC2003 conference in Phoenix (November 16–19). The organization of this project included the creation of teams representing application groups, site administrators, middleware developers, core service providers, and operations. The active period of this project was a 5-month period from July through November 2003. It is interesting to note that subsequent to this period, Grid3 remains largely intact, with many applications running. However, there are reduced expectations as to response time to problems and the attention of the members of the team. Grid2003 was coordinated by the iVDGL and PPDG project coordinators.

The Grid2003 Project deployed, integrated, and operated Grid3 with 27 operational processing sites comprising at peak ~2800 CPUs for more than three weeks. Progress was made in other areas that are important to the iVDGL mission, including the following:

- *Multiple VO grid*: six different virtual organizations participated and successfully deployed 10 applications. All applications were able to run on sites that were not owned by the host organization. Further, the applications were all able to run on non-dedicated resources.
- *Multi-disciplinary grid*: during the project, two new applications, the *SnB* structural biology application and an application in chemical informatics, were run across Grid3. The fact that these could be installed and run on a Grid infrastructure designed and installed for Particle and Astrophysics Experiments provides the members of iVDGL with confidence that this grid can be adapted to other applications as needed.
- *Use of shared resources*: many of the resources brought into the Grid3 environment were leveraged facilities in use by other VOs.
- *Dynamic resource allocation*: in addition to resources that were committed 24×7, the University at Buffalo's Center for Computational Research (CCR) configured their local schedulers to bring additional resources in to and out of Grid3 on a daily basis, satisfying local requirements and Grid3 users.
- *International connectivity*: one site was located abroad (Kyunpook National University, Korea).

Over the course of several weeks surrounding SC2003, the Grid2003 project met its target goals, as follows:

1. *Number of CPUS*. Target: 400, status: 2163. More than 60% of available CPU resources are non-dedicated facilities. The Grid3 environment effectively shared resources not directly owned by the participating experiments.
2. *Number of users*. Target: 10, status: 102. About 10% of the users are application administrators who do the majority of the job submissions. However, more than 102 users are authorized to use the resources through their respective VO'S services.
3. *Number of applications*. Target: >4, status: 10. Seven scientific applications, including at least one from each of the five GriPhyN-iVDGL-PPDG participating experiments, *SnB* structural biology, and GADU/Gnare genome analysis, were and continue to run on Grid3.
4. *Number of sites running Concurrent Applications*. Target: >10 status: 17. This number is related to the number of Computational Service sites defined on the catalog page and varies with the application.
5. *Data transfers per day*. Target: 2–3 TB, status: 4 TB. This metric was met with the aid of the GridFTP-demo.
6. *Percentage of resources used*. Target: 90%, status: 40–70%.
7. *Peak number of concurrent jobs*. Target: 1000; status: 1100. On 20 November 2003 there were sustained periods when over 1100 jobs ran simultaneously.

6. Final remarks

In this paper, we discuss the design and implementation of the Buffalo-based ACDC Grid. We also discuss the implementation of *SnB* and it's ability to perform data management in order to enable data mining on both the ACDC Grid and on Grid2003.

The extremely heterogeneous ACDC Grid is based on Globus and uses NWS, the Apache HTTP Server, PHP, JavaScript, and MySQL, to name a few. The development of the ACDC Grid took approximately 20 staff months, and we project that maintenance of the grid will require 2 FTEs.

We introduced our web portal and lightweight-monitoring tool, which were used in Grid2003. We discussed some of the tools that we have developed for our complementary data grid, which provide the capability of automatically migrating files and providing users with easy access to file management.

Our current plans call for rolling out the ACDC Grid to the general CCR population by the middle of 2004.

Acknowledgments

Support was provided by the Center for Computational Research, NSF grant ACI-02-04918, and NIH grant GM-46733. The authors would like to thank Paul Avery, Robert Gardner, Ruth Pordes, Cynthia Cornelius, Steve Gallo, Martins

Innus, Adam Koniak, and Jason Rappleye for their contributions. Additional information can be found at http://www.ccr.buffalo.edu/.

References

[1] R. Miller, S.M. Gallo, H.G. Khalak, C.M. Weeks, SnB: crystal structure determination via Shake-and-Bake, J. Appl. Crystallogr. 27 (1994) 613–621.
[2] C.M. Weeks, G.T. DeTitta, H.A. Hauptman, P. Thuman, R. Miller, Structure solution by minimal function phase refinement and Fourier filtering: II. implementation and applications, Acta Cryst. A50 (1994) 210–220.
[3] G.T. DeTitta, C.M. Weeks, P. Thuman, R. Miller, H.A. Hauptman, Structure solution by minimal function phase refinement and Fourier filtering: theoretical basis, Acta Crystallogr. A 50 (1994) 203–210.
[4] H.A. Hauptman, H. Xu, C.M. Weeks, R. Miller, Exponential Shake-and-Bake: theoretical basis and applications, Acta Crystallogr. A 55 (1999) 891–900.
[5] C.M. Weeks, R. Miller, The design and implementation of SnB v2.0, J. Appl. Crystallogr. 32 (1999) 120–124.
[6] C.M. Weeks, R. Miller, Optimizing Shake-and-Bake for proteins, Acta Crystallogr. D 55 (1999) 492–500.
[7] J. Rappleye, M. Innus, C.M. Weeks, R. Miller, SnB v2.2: an example of crystallographic multiprocessing, J. Appl. Crystallogr. 35 (2002) 374–376.
[8] M.L. Green, R. Miller, Grid computing in Buffalo, Annals of the European Academy of Sciences, New York, 2003, pp. 191–218.
[9] H.A. Hauptman, A minimal principle in the phase problem, in: D. Moras, A.D. Podjarny, J.C. Thierry (Eds.), Crystallographic Computing 5: from Chemistry to Biology, IUCr and Oxford University Press, Oxford, 1991, pp. 324–332.
[10] http://www.gridcomputing.com.
[11] M.L. Green, R. Miller, A client-server prototype for application grid-enabling template design, Parallel Processing Letters 14 (2) (2004) 241–253.
[12] http://www.ivdgl.org/.
[13] http://www.ivdgl.org/grid2003.
[14] http://www.hicb.org/glue/glue.htm.
[15] http://www.ppdg.net.
[16] http://eu-datagrid.web.cern.ch/eu-datagrid/.
[17] http://datatag.web.cern.ch/datatag/.
[18] http://www.eu-crossgrid.org/.
[19] http://public.web.cern.ch/public/.
[20] http://www.acdc.ccr.buffalo.edu/.
[21] http://www.cse.buffalo.edu/pub/WWW/faculty/miller/Talks_HTML/SC2003/SC2003-Panel.pdf.
[22] I. Foster, C. Kesselman, The Grid: blueprint for a New Computing Infrastructure, Morgan Kauffman Publishers Inc., San Francisco, 1999.
[23] F. Berman, G. Fox, T. Hey, Grid Computing: Making the Global Infrastructure a Reality, John Wiley and Sons, 2003.
[24] http://www.globus.org/.
[25] http://nws.cs.ucsb.edu/.
[26] http://www.griphyn.org/.