GridX1: A Canadian Particle Physics Grid

A. Agarwal[i], M. Ahmed[c], B.L. Caron[d,e], A. Dimopoulos[i], L.S. Groer[h], R. Haria[c], R. Impey[c],
L. Klektau[i], C. Lindsay[i], G. Mateescu[c], Q. Matthews[i], A. Norton[i], D. Quesnel[a], R. Simmonds[f],
R.J. Sobie[b,i], B. St. Arnaud[a], D.C. Vanderster[i], M. Vetterli[d,g], R. Walker[g], M. Yuen[i]

[a] CANARIE Inc., Ottawa, Ontario, Canada
[b] Institute of Particle Physics of Canada
[c] National Research Council, Ottawa, Ontario, Canada
[d] TRIUMF, Vancouver, British Columbia, Canada
[e] Department of Physics, University of Alberta, Edmonton, Canada
[f] Department of Computer Science, University of Calgary, Calgary, Canada
[g] Department of Physics, Simon Fraser University, Burnaby, British Columbia, Canada
[h] Department of Physics, University of Toronto, Toronto, Ontario, Canada
[i] Department of Physics and Astronomy, University of Victoria, Victoria, British Columbia, Canada

Abstract

GridX1, a Canadian computational Grid, combines the resources of various Canadian research institutes and universities through the Globus Toolkit and the CondorG resource broker (RB). It has been successfully used to run ATLAS and BaBar simulation applications. GridX1 is interfaced to LCG through a RB at the TRIUMF Laboratory (Vancouver), which is an LCG computing element, and ATLAS jobs are routed to Canadian resources. Recently, the BaBar application has also been implemented to run jobs through GridX1, concurrently with ATLAS jobs. Two independent RBs are being used to submit ATLAS and BaBar jobs for an efficient operation of the grid. The status of grid jobs and the resources are monitored using a web-based monitoring system.

INTRODUCTION

GridX1 is a collaborative project that is establishing a computational grid infrastructure across Canada. Canada has a number of medium to large scale computing facilities with a broad range of clusters, parallel, and vector machines. GridX1 unifies some of the cluster resources into a large virtual facility. The resources are not dedicated to GridX1 but instead treat the grid users as another local user. This allows GridX1 to exploit unused cycles at these facilities.

The facilities available to GridX1 are shared with many fields of research. Each facility operates independently with its own management team and user community. GridX1 uses generic middleware to minimize the management requirements without compromising security or interfering with the normal operation. A metascheduler accepts advertisements from computational resources and allocates user jobs to the clusters. Finally, GridX1 features command-line and web-interfaces to the metascheduler and resource monitoring systems.

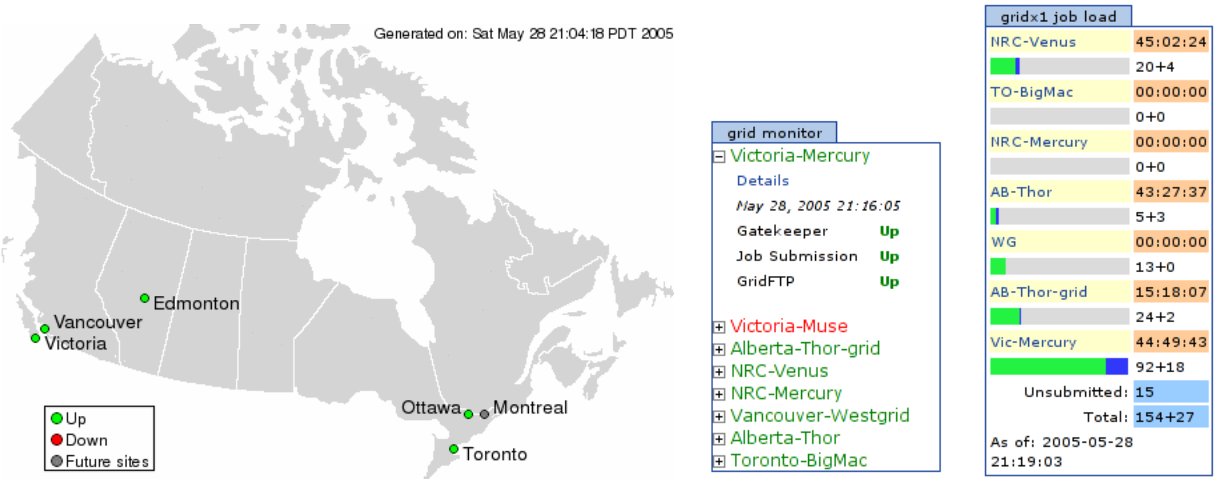
The design of GridX1 has been focused around the execution of embarrassingly-parallel applications, which are typically simulations requiring relatively small amounts of data. The first users of GridX1 have been particle physicists working on the ATLAS experiment for the Large Hadron Collider (LHC) project at CERN and the BaBar experiment at the Stanford Linear Accelerator Center (SLAC). By developing an interface which federates the GridX1 resources into the LCG, we have contributed substantial computing resources to the project while maintaining the shared nature of the facilities. Similarly, a separate resource broker has been deployed to manage the BaBar simulation application.

This paper highlights the GridX1 resources and its infrastructure that are used to execute the ATLAS and BaBar simulation applications.

GRIDX1 RESOURCES

GridX1 resources, presented in Figure 1 (a), include clusters at the Centre for Subatomic Research at the University of Alberta, the Research Computing Centre at the University of Victoria, the WestGrid cluster at the University of British Columbia, the Research Computing Support Group at the National Research Centre in Ottawa, a cluster at McGill University in Montreal, and the BigMac cluster at the University of Toronto High Energy Physics Group. In some instances, sites may have more than one cluster. The total number of processors at these sites is approximately 2500 with disk and tape storage well in excess of 100 TB. GridX1 is given access to a fraction of these resources, with some sites allocating a specific job quota, and others allowing GridX1 to backfill during periods of low utilization.

Each cluster uses 32-bit x86 processors from Intel and AMD ranging in clock speed from 1 to 3.2 GHz. A variety of operating systems are implemented, including RedHat



grid monitor

⊟ Victoria-Mercury
Details
May 28, 2005 21:16:05
Gatekeeper Up
Job Submission Up
GridFTP Up

⊞ Victoria-Muse
⊞ Alberta-Thor-grid
⊞ NRC-Venus
⊞ NRC-Mercury
⊞ Vancouver-Westgrid
⊞ Alberta-Thor
⊞ Toronto-BigMac

gridx1 job load

NRC-Venus	45:02:24
	20+4
TO-BigMac	00:00:00
	0+0
NRC-Mercury	00:00:00
	0+0
AB-Thor	43:27:37
	5+3
WG	00:00:00
	13+0
AB-Thor-grid	15:18:07
	24+2
Vic-Mercury	44:49:43
	92+18
Unsubmitted:	15
Total:	154+27

As of: 2005-05-28
21:19:03

(a) (b) (c)

Figure 1: A web-based monitoring system presents the current resource statuses: (a) a summary map with resource location and status of seven clusters in five cities, (b) the detailed middleware status for each cluster, and (c) the number of jobs running, waiting, and the estimated wait time at each cluster.

Enterprise Linux 3, Scientific Linux (RHEL3-compatible), and SUSE Linux. Most sites have deployed the Portable Batch Scheduler (PBS) [1, 2] software for local resource management, with an exceptional few deploying the Condor batch system [3]. The sites are configured with one or more head nodes which act as a user interface to the worker nodes. The worker nodes are required to have external network access via network address translation or a similar technique. Most of the sites have 1 gb/s network connectivity to the national research network provided by CANARIE.

THE GRIDX1 INFRASTRUCTURE

GridX1 has been deployed using version 2 of the Globus Toolkit [4] as packaged in the Virtual Data Toolkit [5]. This version of the middleware is popular with production grid deployments due to its maturity, which has resulted in a stable deployment platform. In addition to the basic middleware, a number of GridX1-specific tools have been developed to manage users and monitor tasks and resources.

Security and User Management

GridX1 employs the standard Grid Security Infrastructure [6], implemented in the Globus Toolkit, to provide single-sign-on authentication, authorization mechanisms, secure communication, and the auditing of user actions. All GridX1 hosts and users are assigned an X.509 certificate [7] by the Grid Canada Certificate Authority (CA), an internationally recognized CA. Further, the GridX1 hosts recognize certificates signed by CAs from around the world.

GridX1 has adopted a simple, however manual, user management system. The master list of GridX1 users is maintained centrally. Site administrators are encouraged to implement local user accounts following a standard naming system of the form gcprod$x$, where x is an account number. To allow for flexibility at the resources, each site is free to apply extensions to the user mapping table to accommodate non-standard users or to remove entries to prevent access by specific users. Additionally, GridX1 has deployed the MyProxy [8] online credential repository. This allows users to obtain credentials when they are needed from any host supporting the MyProxy client.

Resource Management and Metascheduling

GridX1 utilizes the Condor-G [9] resource management system for cluster metascheduling. The Condor system provides great flexibility in interfacing users to the GridX1 resources. Condor separates the processes which handle the management of resource advertisements, task queues, and matchmaking between tasks and resources. Additionally, Condor does not restrict system designers to a one-to-one mapping between the processes. GridX1 has made use of this flexibility to provide for a scalable metascheduling architecture.

Figure 2 presents the metascheduling architecture of GridX1 for both BaBar (top) and ATLAS (bottom). Users access the GridX1 metascheduler via a GRAM or Condor interface. Each of the interfaces is associated with a task queue. Users may use the official GridX1 interfaces or deploy a desktop interface. Cluster resources publish Classified Advertisements (ClassAds) to a Condor collector. The resource ClassAd is generated by a customized script running from a non-privileged crontab process on the head node of each cluster. Each of the interfaces is configured to access a collector and a matchmaking service (also known as a negotiator). The matchmaking service matches each task to an appropriate cluster resource. The flexible nature of Condor allows GridX1 to have multiple schedulers, collectors, and negotiators. By dividing the task workload

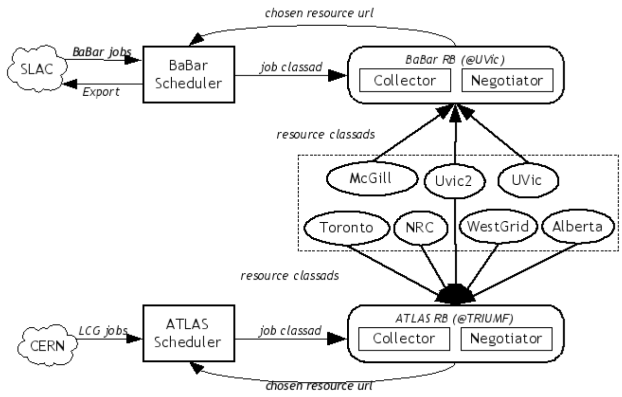


Figure 2: The GridX1 metascheduling architecture allows users to submit via GRAM or Condor for BaBar (top) and ATLAS (bottom). Cluster resources publish ClassAds to a Condor collector. The Condor negotiator matches the task and resource ClassAds to choose appropriate resources.

between interfaces, GridX1 can scale to thousands of simultaneous tasks.

Monitoring and Operations

The operation of GridX1 relies on the availability of current, accurate, and easily accessible status information. By employing a web-based grid monitoring system, users can easily track the status of resources and their tasks, and additionally, the grid operators can detect and diagnose faults in the grid.

The GridX1 monitoring system is built around a central daemon which periodically evaluates the health of the grid. The middleware status of each cluster resource is determined by performing an authentication test, a GRAM job submission test, and data transfer test. The results of these tests are archived to a MySQL database.

A web-interface to the monitoring system presents status information in an easily accessible form. Overall grid health is summarized in a national map of cluster resources (depicted in Figure 1 (a)). Detailed middleware statuses are also presented (as in Figure 1 (b)). Additionally, web-interfaces to the metascheduling system allow users to monitor resource and task statuses. Figure 1 (c) shows a bar chart which tracks the resource usage and availability at each cluster as advertised to the resource registry. The current status of each task queue is displayed in table form (not shown here). Cluster-level monitoring is provided by Ganglia [10] at most of the sites.

USER APPLICATIONS ON GRIDX1

ATLAS and the LCG

GridX1 has been used extensively for the ATLAS particle physics simulation application, notably during the 2004 Data Challenge 2 (DC2) [11] and the 2005 Rome Production. In order to make the GridX1 resources available to the LCG, an interface which implemented the virtualization layer was developed [12]. The interface to LCG is provided by an LCG Compute Element, with a Globus JobManager of type Condor-G (provided by condorg.pm) to interface to the GridX1 resource management system. Also provided is an information provider which advertises the combined availability of all the GridX1 resources to the LCG. The Condor-G JobManager proceeds by creating a Condor-G job description file, submitting to the Condor scheduler using condor_submit, and polling the job using condor_q. These actions act as if the resources were in a local cluster.

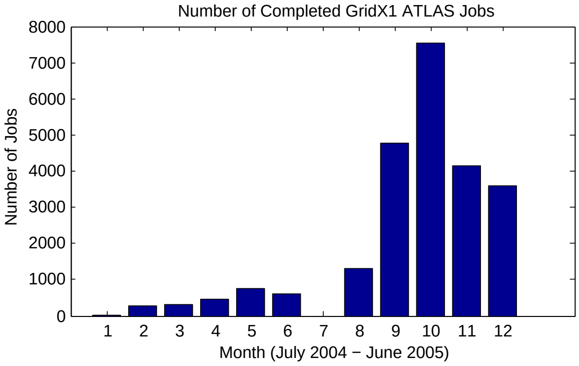


Figure 3: Plot of ATLAS jobs successfully executed on GridX1 over the period starting July 2004 and ending June 2005. GridX1 has been used to execute over 20000 ATLAS jobs.

Since the re-submission to the GridX1 clusters proceeds via the GRAM protocol, it requires a *full user proxy*. This is obtained by utilizing the MyProxy online credential repository. This service is used by the Condor-G JobManager to delegate a full proxy using a limited proxy to authenticate.

Over the course of the ATLAS DC2 and Rome productions, the GridX1 system has been quite effective in both execution efficiency and ease of maintenance. As depicted in Figure 3, the number of successful ATLAS jobs executed on GridX1 exceeds 20000. Additionally, the success rate of jobs on GridX1 has been similar to that of the entire LCG at approximately 50%. This seemingly large failure rate is due to instabilities in the LCG as a whole (90% of the failures were due to the unavailability of a few storage elements), and rarely did GridX1 specific issues cause problems. Cluster utilization has remained high, with usage loads peaking at over 200 simultaneously running jobs.

BaBar Monte Carlo

The BaBar experiment at the Stanford Linear Accelerator Center studies the asymmetry between matter and antimatter in the Universe. In parallel with the physical detector, the BaBar Monte Carlo (MC) application generates synthetic particle collisions, which are used for consistency checking and physics analysis. The Canadian computational contribution to the BaBar experiment has typically

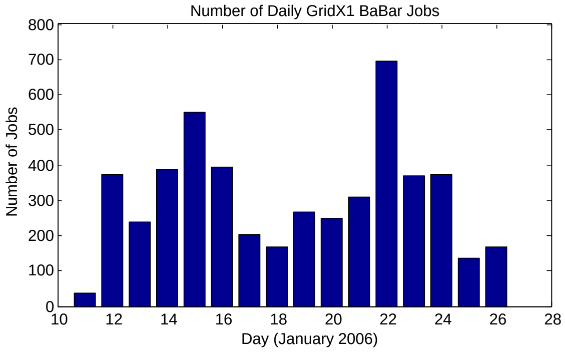


Figure 4: Plot of BaBar jobs successfully executed on GridX1 over the period of January 11-26, 2006.

come in the form of a number of individually managed cluster resources. Recently, BaBar MC has been adapted to execute on GridX1. As a result, the application management is simplified by collapsing multiple managers to a single interface. Further, utilizing GridX1 provides the application with access to new resources which will result in increased production rates.

The process of grid-enabling the BaBar MC requires the installation of the application package at each GridX1 resource. This includes a local Objectivity database server containing a conditions database and background triggers. The metascheduling architecture is illustrated in Figure 2 (top). A submit workstation, holding the BaBar scheduler, is configured to act as the BaBar MC grid management node using Condor-G, and it handles application specific tasks such as building and creating tar archives of the input execution directories. When a job is submitted to the GridX1 RB at UVic, the matchmaking process selects an appropriate resource for the execution of the job. Once the job is finished, a tar archive of the output is transferred back to the submit workstation. The output data is merged and exported to the production database at SLAC. A web-based monitor presents the status of the BaBar production on GridX1. A plot of BaBar jobs successfully executed on GridX1 over the period of January 11-26, 2006 is exhibited in Figure 4. As noted from the figure, 200 or more BaBar jobs are running successfully daily on the UVic and McGill clusters. Soon we will be adding more cluster resources to enhance the daily MC production.

The results of this activity have been successful for the BaBar community. The approach presents a computing solution that allows the production rates to increase without substantial increases in administration costs.

CONCLUSIONS

The architecture of GridX1 emphasizes standards compliance, by building around the de-facto standard Globus Toolkit, and ease-of-deployment, by leveraging proven technologies such as the Virtual Data Toolkit and Condor-G. The incorporation of Condor's matchmaking system into the GridX1 metascheduler has allowed for reliable operation and provided a mechanism for improving the resource utilization. Web-based monitoring information has proven to be useful to grid operators and users alike. By exploiting the GridX1 infrastructure and resources, both ATLAS and BaBar simulation applications proved to be highly efficient and successful.

ACKNOWLEDGEMENTS

The authors would like to thank the Grid Canada collaborators and acknowledge the support of the Natural Sciences and Engineering Research Council of Canada, and the National Research Council of Canada. We also acknowledge the contributions of the GridX1 local resource sites.

REFERENCES

[1] Portable Batch System Professional (2005).
URL http://www.altair.com/pbspro/

[2] OpenPBS (2005).
URL http://www.openpbs.org/

[3] T. Tannenbaum, D. Wright, K. Miller, M. Livny, Condor – A Distributed Job Scheduler, in: T. Sterling (Ed.), Beowulf Cluster Computing with Linux, MIT Press, 2001.

[4] I. Foster, C. Kesselman, Globus: A Metacomputing Infrastructure Toolkit, International Journal of Supercomputer Applications 11(2) (1997) 115.

[5] Virtual Data Toolkit (2005).
URL http://www.cs.wisc.edu/vdt/

[6] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, A Security Architecture for Computational Grids, in: Proceedings of the 5th ACM Conference on Computer and Communications Security Conference, 1998, p. 83.

[7] R. Housley, W. Ford, W. Polk, D. Solo, Internet X.509 Public Key Infrastructure, RFC 2459 (Jan. 1999).

[8] J. Novotny, S. Tuecke, V. Welch, An Online Credential Repository for the Grid: MyProxy, in: Proceedings of the Tenth International Symposium on High Performance Distributed Computing, IEEE Press, 2001.

[9] J. Frey, T. Tannenbaum, M. Livny, I. Foster, S. Tuecke, Condor-G: A Computation Management Agent for Multi-Institutional Grids, in: Proceedings of the Tenth International Symposium on High Performance Distributed Computing, IEEE Press, 2001.

[10] M. L. Massie, B. N. Chun, D. E. Culler, The Ganglia Distributed Monitoring System: Design, Implementation, and Experience, Parallel Computing 30 (2004) 817.

[11] The ATLAS Data Challenge (2005).
URL http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/DC/

[12] R. Walker, M. Vetterli, R. Impey, G. Mateescu, B. Caron, A. Agarwal, A. Dimopoulos, L. Klektau, C. Lindsay, R. J. Sobie, D. Vanderster, Federating Grids: LCG Meets Canadian HEPGrid, in: Proceedings of Computing in High Energy and Nuclear Physics 2004, 2004.