FEDERATING GRIDS: LCG MEETS CANADIAN HEPGRID

R. Walker, M. Vetterli*, Simon Fraser University, Vancouver, British Columbia, Canada

R. Impey, G. Mateescu, NRC Institute for Information Technology, Ottawa, Ontario, Canada

B. Caron†, University of Alberta, Edmonton, Alberta, Canada

A. Agarwal, A. Dimopoulos, L.Klektau, C. Lindsay, R.J. Sobie‡, D. Vanderster, University of Victoria, Victoria, British Columbia, Canada

Abstract

A large number of Grids have been developed worldwide. Despite being mostly based on the same underlying middleware, the Globus Toolkit, they are generally not inter-operable for a variety of reasons. We present a method of federating those disparate grids which are based on the Globus Toolkit, together with a concrete example of interfacing the LHC Computing Grid (LCG) with HEPGrid. HEPGrid consists of shared resources, at several Canadian research institutes, which are exposed via Globus gatekeepers, and makes use of Condor-G for resource advertisement, matchmaking and job submission. An LCG Computing Element (CE) based at the TRIUMF Laboratory hosts a HEPGrid User Interface (UI) that is contained within a custom JobManager. This JobManager appears in the LCG information system as a normal CE publishing an aggregation of the HEPGrid resources. The interface interprets the incoming job in terms of HEPGrid UI usage, submits it onto HEPGrid, and implements the JobManager 'poll' and 'remove' methods, thus enabling monitoring and control across the grids. In this way non-LCG resources are integrated into LCG, without the need for LCG middleware on those resources. The same method can be used to create interfaces between other grids, with the details of the child-Grid being fully abstracted into the interface layer. The LCG-HEPGrid interface is operational, and has been used to federate 1300 CPU's at 4 sites into LCG for the ATLAS Data Challenge (DC2).

INTRODUCTION

The LHC Computing Grid (LCG) [1] is being developed to analyse the enormous amount of data that will be generated by the LHC experiments starting in 2007. Canadian physicists are participating in one of these experiments, ATLAS. The ATLAS-Canada computing model is centred on a large-scale computing and storage facility at TRIUMF for common computing tasks, complemented by significant resources in the universities for physics analysis and Monte Carlo simulation. The latter facilities are shared with other fields of research that have their own requirements for computing and networking. As a result, the Canadian model is based on a national Grid that uses as generic middleware as possible to minimize interference with other disciplines. This national Grid is interfaced to LCG through the TRIUMF centre. This allows for simpler management of the shared university centres and for Canadian control of the load balancing, both across the Canadian network, and between this Grid and the LCG. This paper presents two aspects of the Canadian model: the design of the Canadian Grid, and of the interface between it and the LCG.

BUILDING A GRID WITH CONDOR-G

In order to federate non-LCG Canadian resources into LCG, these resources must first be part of a Grid themselves, or at least a coordinated distributed computing environment. This was achieved by the use of Globus gatekeepers [2], and Condor-G [3]. With remarkably little development work, these two components can form a fully functional distributed computing environment, essentially out-of-the-box. The deployment of Condor-G in this way [4] is sufficiently novel to warrant discussion here.

Information System

The Condor ClassAd mechanism is used to advertise resource characteristics and status to central *Collectors*. This is the identical mechanism used within a plain Condor batch system. The ClassAd consists of attribute-value pairs, where the attributes can be static or dynamically probed from the Local Resource Management System (LRMS). There are several mandatory attributes in order for Condor to recognize the resource as a Globus-enabled cluster, rather than a single execution host (as in the plain Condor LRMS case). The

* Also TRIUMF, Vancouver, British Columbia, Canada

† Also TRIUMF

‡ Also Institute of Particle Physics

remainder of the ClassAd is free format, i.e. there is no rigid schema. This allows great freedom to fully represent the cluster, providing the format is well documented for the users.

We have developed a script to probe the LRMS, produce the ClassAd and advertise it to the *Collectors*. This is installed on each resource and runs from a non-privileged crontab process.

Resource Brokerage

The Condor *Negotiator* is responsible for matching jobs to resources. It does this by periodically taking job ClassAds and resource ClassAds from the *Collector*, and using the *Rank* and *Requirements* therein to choose the best resource. The *Requirements* form a logical expression which in the case of the job ClassAd describes the user's requirements, e.g. OS, memory, wall-time. The *Requirements* would also be used to ensure the user is authorized to use a particular resource. In the case of the resource ClassAd this could express the site requirements, e.g. no jobs accepted during office hours, or setting to FALSE to prevent any job matching there at all. Both the job and resource *Requirement* expressions must evaluate to TRUE for the resource to be considered in the next stage of match-making.

Then the *Rank* expression is evaluated to a floating-point number for each job-resource combination. The pair with the highest *Rank* is matched, and the job will go to this resource. The *Rank* is used to express user and resource preferences. A typical user *Rank* would be the negative estimated waiting time until the job runs. This would ensure the job starts running as soon as possible. The CPU SI2k rating or the available RAM might also be used in the user *Rank*. The resource *Rank* is less obvious, but may be used to prefer certain user groups, or maybe discourage big memory usage.

INTERFACE TO LCG

The LCG Computing Element (CE) consists of a Globus gatekeeper, with a JobManager to interface to the Local Resource Management System (LRMS), and an information provider. The LRMS is most often PBS, although other batch systems are supported, e.g. the plain Condor batch system. The information provider probes the LRMS to provide the queue status in MDS GLUE [5] format.

In order to use Condor-G as our LRMS, we must provide both the Condor-G JobManager, and the information provider. In both cases, this is greatly helped by the existence of the plain Condor equivalents.

The Condor-G client allows us to submit and monitor jobs just as if they were on a local batch system, but there is one important difference – the submission to Grid-Canada (GC) nodes proceeds via the GRAM protocol and hence a 'full' user proxy is required. The proxy that arrives with the job, from the LCG Resource Broker (RB), is 'limited', i.e. it can be used for gridftp transfers, but not for a further GRAM submission. Full proxy delegation via the GRAM protocol is possible, however this may be a security concern. We also considered using a shared proxy that can be picked up from the interface machine, but then accountability would be lost. In any case there is a straightforward solution, making use of the LCG proxy renewal service.

How to get a 'Full' proxy

The LCG proxy renewal service ensures that a user's proxy does not run out during a long Grid job. The user first stores his full proxy in a MyProxy [6] server. Normally the LCG RB would use the user's proxy, which is about to expire, to authorize the delegation of a new one from the MyProxy server. The same thing can be done on the interface machine, but unlike the RB we have no way of knowing which MyProxy server was used. Inside the Condor-G JobManager, we search known MyProxy servers until the correct one is found. This is not satisfactory, and is the result of too little information being passed with the LCG job – a recurring theme.

Building the Condor-G job description file

As with several batch systems, and LCG itself, a Condor-G job is described in a job description file (JDF). This is prepared by the JobManager, using information in the GRAM Resource Specification Language (RSL) [7]. The RSL attributes are rather limited, but the required wall-time, cpu-time, and memory are provided. Unfortunately LCG chooses not to set these attributes, and assumes that the entire LRMS resource is characterised by the information published in MDS. This is often not the case for current farms, where a variety of CPU speeds and memory sizes may exist due to partial upgrades. In our case, the LRMS is itself a heterogeneous sub-Grid and we are particularly affected by this lack of information.

In fact, many batch systems, and certainly Condor-G, can deal with a wide-range of job requirements and ensure that the best worker node (WN) is selected. For example, OS, experiment software, and WN disk-space could all be accommodated, were they available to the JobManager, and hence JDF.

In the controlled environment of ATLAS DC2 it was possible to enforce homogeneity on the different clusters. In general, this will not be possible and some mechanism to pass the job requirements to the LRMS will be necessary.

Requirements on the Worker Nodes

The LCG job arriving at the GC interface is simply passed on to GC without modification. In fact, it would be possible do to many things to make the job suitable for the sub-Grid. We chose not to change the job, but rather make the WNs look and feel like genuine LCG WNs. The principal attributes the LCG expects are:

- ***Outbound IP connectivity***
- ***Gridftp client and trusted Certificate Authorities***
- ***LCG data handling tools***

- ***Experiment software***

Each of these could be satisfied in a non-intrusive way by a combination of reasonable requests to the system administrators, and the use of a shared NFS area managed by a non-privileged user. The lack of a data handling system on Grid-Canada led us to use the LCG tools, which were found to be quite portable.

PERFORMANCE

It was found that the bug fixes and enhancements to Globus 2.4.3, which are in the VDT [8] packaged software, were necessary to have the system scale. In particular, the grid-monitor tool provided by CondorG, and the associated Globus hooks, were essential to keep the load down on the gatekeeper machines. Many of these fixes were discovered and implemented by the LCG team, which highlights the degree of technology re-use.

In practise, this system has been quite effective while running the ATLAS DC2. Over the course of DC2, the success/failure ratio of jobs on HEPGrid has been similar to that on the entire LCG. The flexibility of the system has been demonstrated when problems have arisen at a specific cluster. In this situation, the Requirements expression is easily modified to exclude the problematic cluster from the matchmaking process.

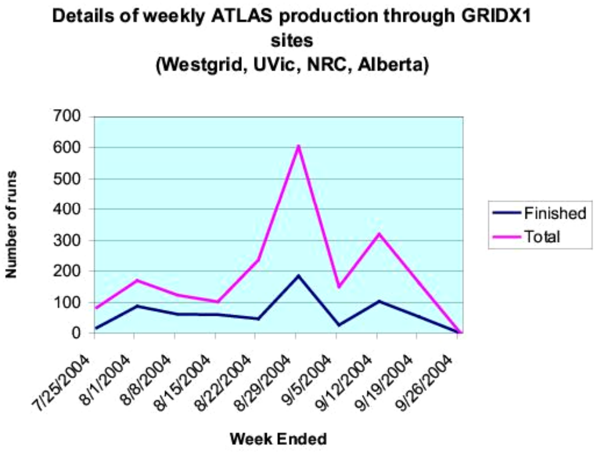

Fig.1: Performance of Grid-Canada interfaced to LCG through TRIUMF. The success/failure rate is consistent with that seen on regular LCG nodes.

CONCLUSIONS

We have demonstrated a working solution to the problem of federating Grids. This has made available to the ATLAS DC2 some 1300 CPUs at 4 sites in Canada, interfaced through the TRIUMF LCG node. These sites were otherwise unable to contribute resources to LCG. This was achieved by combining and deploying a number of off-the-shelf tools with a small amount of development work. Preparing a resource to run LCG jobs in this way is much easier than becoming an LCG site, and was achieved with minimal manpower. In production use, the system has successfully executed ATLAS DC2 jobs, initially submitted to LCG, on Canadian HEPGrid resources.

In addition to its lightweight installation and the resulting need for only modest manpower, this solution has the advantage of providing load balancing on the child-Grid at the interface machine and therefore under the control of "local" administrators. Furthermore the middleware on the child-Grid does not have to be upgraded whenever there is an upgrade of LCG. Only the interface machine needs to have the latest LCG release, aside from details of the data handling. This greatly simplifies the management of the shared facilities on Grid-Canada.

Further work will incorporate improved measures of cluster performance and error handling into the resource brokering process. Deployment of a data handling mechanism will make this Condor-G based system a fully functional Grid on its own. Finally, we plan to move some of the Canadian resources currently configured as separate LCG sites into the Grid-Canada/LCG interface mode, perhaps for the last phase of DC2.

ACKNOWLEDGEMENTS

The modifications to Condor-G to enable Grid resource brokerage are the result of collaboration between the SAMGrid (Dzero) and Condor teams. Clearly the work relies heavily on the Condor software, and we thank them for many useful private communications. We also thank the Grid Canada collaborators and acknowledge the support of the Natural Sciences and Engineering Research Council of Canada, the National Research Council of Canada, WestGrid, and CANARIE Inc.

REFERENCES

[1] LHC Computing Grid Project, see http://lcg.web.cern.ch/LCG/.

[2] Globus Toolkit, see http://www.globus.org/.

[3] J. Frey, T. Tannenbaum, M. Livny, I. Foster, S. Tuecke, "Condor-G: A Computation Management Agent for Multi-institutional Grids" in Proceedings of 10-th International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, July 2001, San-Fransisco, CA

[4] I. Terekhov, G. Garzoglio, A. Baranovskii, S. Patil, A. Rana, H. Kouteniemi, L. Lueking, R. Walker, A. Roy, T. Tannebaum, "Grid Job and Information Management for the FNAL Run II Experiments", CHEP 2003, La Jolla, California, March 2003.

[5] GLUE Schema, see http://www.cnaf.infn.it/~sergio/datatag/glue/

[6] J. Novotny, S. Tuecke, and V. Welch, "An Online Credential Repository for the Grid: MyProxy". Proceedings of the Tenth International Symposium

on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.

[7] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, S. Tuecke. "A Resource Management Architecture for Metacomputing Systems". Proceedings of IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing, pg. 62-82, 1998.

[8] Virtual Data Toolkit, see http://www.cs.wisc.edu/vdt/.

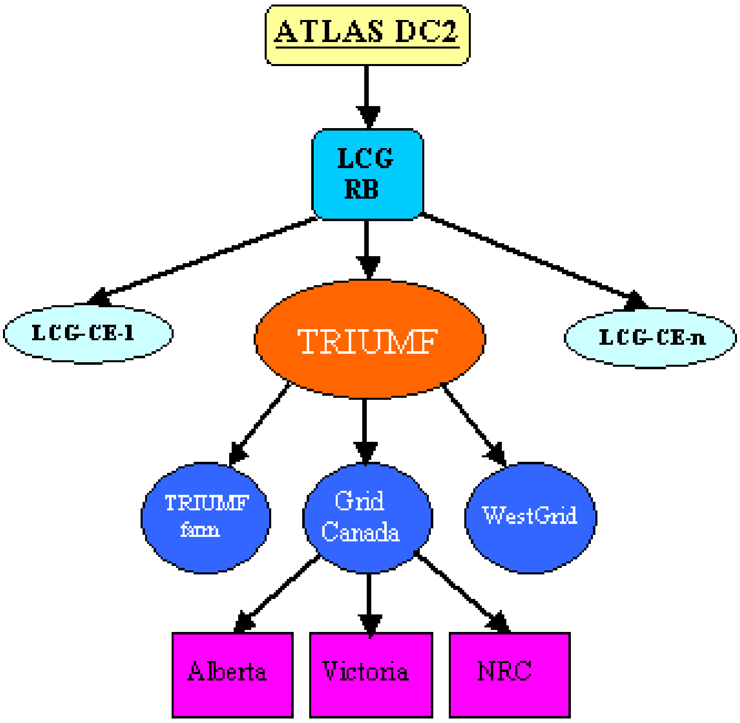

Fig.2: A schematic of the Canadian model for participation in ATLAS DC2. Non-LCG resources were incorporated using a CE at TRIUMF running Condor-G, which acts as a gateway to Grid Canada.