University of Victoria
Faculty of Engineering
Summer 2007 Work Term Report

Developing a Monitoring Database for a Computing Grid

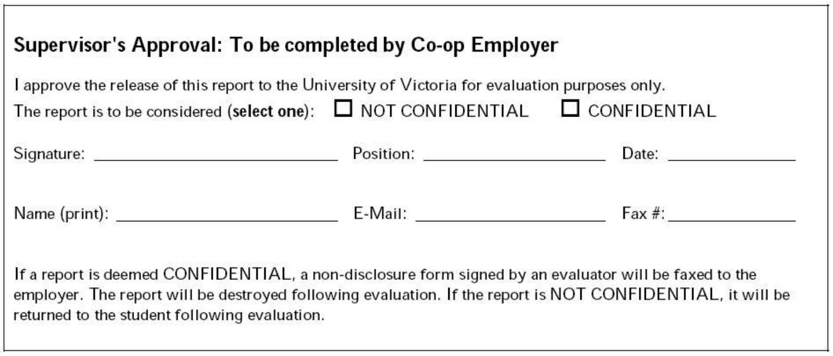
Department of Physics and Astronomy
University of Victoria
Victoria, British Columbia

Sydney Schaffer
03 31455
Work Term 3
Computer Science/Mathematics
schaffer@uvic.ca

29 August, 2007

In partial fulfillment of the requirements of the
Bachelor of Science Degree

Abstract



Supervisor's Approval: To be completed by Co-op Employer

I approve the release of this report to the University of Victoria for evaluation purposes only.
The report is to be considered (**select one**): □ NOT CONFIDENTIAL □ CONFIDENTIAL

Signature: ______________________ Position: __________________ Date: ____________

Name (print): ____________________ E-Mail: __________________ Fax #: ____________

If a report is deemed CONFIDENTIAL, a non-disclosure form signed by an evaluator will be faxed to the employer. The report will be destroyed following evaluation. If the report is NOT CONFIDENTIAL, it will be returned to the student following evaluation.

The University of Victoria (UVic) Department of Physics and Astronomy has developed a computing test grid consisting of a Condor metascheduler, a central registry, and two clusters, known as Fate and Mercury, for high energy physics applications. The grid has been used for submission of test jobs. Summarizing the history of jobs which have already been completed (through either success or failure) is a computationally-intensive task, which cannot itself be submitted to the grid. The existing monitoring script was based on the Condor Scheduler, which poses serious limitations due to large numbers of job entries in it. To improve performance of this task, a separate PostgreSQL database, maintained by Condor's Quill daemon, is created to store all previous job information. This makes the job monitoring process faster and frees up the main scheduler program to perform its primary function more efficiently.

Report Specification

- **Audience** – This report is written for the benefit of later students in my position, who will be working on High Energy Physics applications, as well as the coordinators at the UVic Co-op program.
- **Prerequisites** – An understanding of computer networking concepts, while not entirely necessary, would be a great help to a potential reader of this report.
- **Purpose** – This work is primarily designed as a reference documenting the work I have done, to help later students in my position. Though the systems have now been set up, descriptions of the methods used should be helpful to anyone who needs to deal with them in the future.

Table of Contents

List of Figures

Glossary

- CANARIE: The Canadian Advanced Network and Research for Industry and Education. CANARIE's mission is to accelerate Canada's advanced Internet development and use by facilitating the widespread adoption of faster, more efficient networks and by enabling the next generation of advanced products, applications and services to run on them [CANARIE].
- Cluster: A group of computers consisting of one head node and at least two (often several hundred) worker nodes. All members of a cluster are generally near each other physically.
- Condor: A workload management system designed with computing grids in mind. GridX1 uses Condor version 6.8.5 to manage its jobs.
- Daemon: A computer program which is launched by another program to perform a specific task: for example, reporting the status of other computers in the grid.
- Grid: A network of computers consisting of at least one metascheduler and at least two clusters. Unlike in a cluster, all the components of a grid are generally not geographically near to each other.
- GridX1: A collaboration between UVic, the NRC, and the CANARIE project to develop software for computational grids.
- Job: A program submitted to a grid through a metascheduler. The job may be transmitted to the worker node when it is submitted, or may already be present on the

worker node.

- Metascheduler: One of the primary computers in a grid, a metascheduler collects information on all clusters in a grid and distributes jobs to the most appropriate clusters.
- Node: Any computer in a grid can be referred to as a node.
- NRC: The National Research Council of Canada.
- Queue: On a metascheduler or head node, the queue is the dynamically-updated list of jobs which have been submitted and their current status. Jobs which are running on worker nodes are in the queue, as are jobs which have not yet been sent to a worker node, but completed jobs are removed from the queue.
- Quill: A daemon, included in the Condor distribution, the primary purpose of which is to scan the job queue and recent job history and copy this information into a PostgreSQL database. A secondary purpose of Quill is to format and report queries to the queue and history, relieving the Scheduler daemon of this requirement and the user of the need to understand SQL queries.
- PBS: Portable Batch System. A job submission system used by the test grid.
- PostgreSQL: An open-source implementation of SQL, a database management system.
- Registry: A machine which centralizes and stores the information of all the machines in a grid. Only one registry is needed in any grid.
- Resource Head Node: The "main" computer in a cluster, which acts as an intermediary between worker nodes and the metascheduler, receiving jobs from the metascheduler and

choosing which worker node would be most appropriate to run them.

- Scheduler: A daemon, included in Condor, which is primarily responsible for organizing jobs submitted, and assigning them to the most appropriate node. When Quill is not configured, its secondary purposes include reporting the queue and history to users, which can be a very slow process when either is large.
- Submission: The act of telling the metascheduler to run a job.
- Worker Node: One of the computers on the grid whose job is to actually execute jobs which have been submitted.

1. Introduction

For several years now, the University of Victoria Department of Physics and Astronomy has been developing a computing grid, a network of computers which can be used to study physics problems for the University's High Energy Physics projects. As physics research becomes more and more complicated, the complexity of the calculations required increases exponentially, to the point where a single computer can not hope to complete the tasks in a reasonable amount of time.

This report will describe the calibration of a program to monitor the jobs running on the many computers connected by the grid. It will first explain the concept of the grid in detail, then the way the job submission system was monitored before this work term began. It will go on to describe the methods used in setting up the new system, then to describe the state of the grid at the time of this report's presentation.

This work term included many small projects, and this report will not attempt to describe them all. Instead, this report will focus only on the largest project completed during the work term: that of improving the methods used to monitor the jobs submitted to the grid.

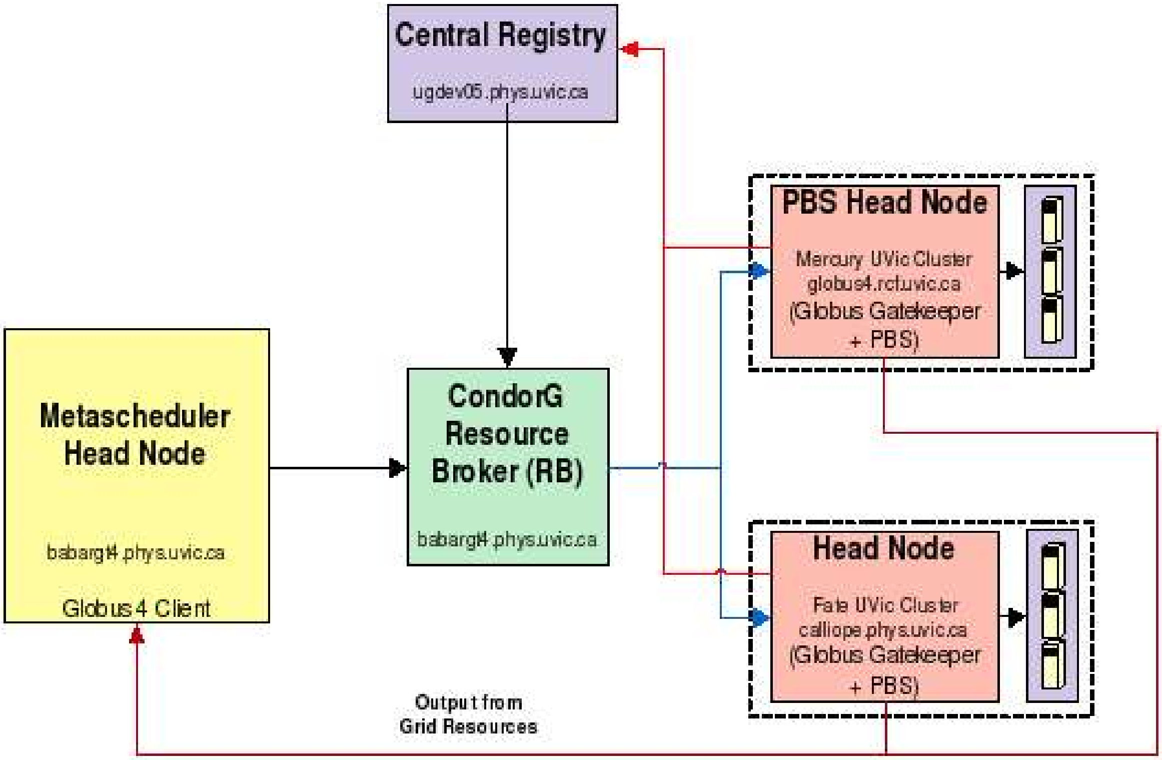
2. A Brief Description of Grid Computing and Condor

Before we can discuss the monitoring of jobs on a computing grid, it would be useful to define the concept. A computing grid is a large network of computers, to which computationally-intensive programs may be submitted as “jobs”. A grid can be viewed as a large network of computing clusters.

A grid may have several machines which are designated “metaschedulers”. These machines do not normally run jobs themselves, but pass jobs on to machines designated as “head nodes” for each cluster. These head nodes, in turn, send jobs to “worker nodes”, which actually run the jobs. A submitter may specify as a requirement that only a particular machine may run a given job, or may leave out this requirement and allow the metascheduler to choose the most appropriate machine.

The information on each machine, including connection information, current status, and local batch systems, is stored in a separate machine called the “registry”. Each resource head node periodically submits its status to the registry, and metaschedulers download this information as part of the job match-making process.

Figure 1: The network setup for UVic's test bed. All machines are located at UVic.



The Condor project is a batch control system for grid computing networks, developed by the University of Wisconsin [Condor]. It is designed as a set of separate programs ("daemons"), not all of which are necessary on any given part of the grid. For example, a metascheduler will need to run the Scheduler daemon, but a worker node will not, while the node will need the Starter daemon, which is unnecessary on the metascheduler.

The UVic Department of Physics and Astronomy has set up a grid test bed by integrating the resources of two computing clusters, known as the "Fate" and "Mercury" clusters, with a metascheduler (babargt4) and registry (ugdev05) developed for this purpose. Figure 1 illustrates the setup of the grid.

3. Existing Grid Job Monitoring Script Using Condor History

At the beginning of this work term, monitoring of grid jobs was performed by a program which made a series of requests to the Scheduler daemon. This program performs archival functions for a metascheduler, but its primary purpose is to control the job queue, sending jobs to head nodes and handling errors in execution. Having the Scheduler perform both functions was fine when the system was first being set up, but as the metascheduler was expected to eventually handle hundreds of job submissions at a time, this could not be a viable solution forever.

The monitoring itself was done by first generating arrays from the information presented in a nonspecific query to the Scheduler, throwing away all information more than two weeks old, and then making specific queries regarding each individual job run in the past two weeks. This information was presented in several graphs, one for each head node, in HTML format, and placed on the Web for easy access.

When the job history became sufficiently large, this became a slow and tedious task. Running the original monitoring script on a history of just three thousand items took three hours to complete, and while it was running, all other requests to the scheduler were slowed as well, though not nearly to that degree. This is a serious limitation and could not be a permanent solution.

4. Setup and Testing of the Condor Quill

The Condor project has developed a daemon called Quill, the purpose of which is to monitor the submission of jobs to Condor and copy all information from the Scheduler into a PostgreSQL database. When the Quill daemon is operational, all queries to Condor's History will not go to the Scheduler daemon (unless the user requests it specifically) but to the Quill, which will respond with the information from the database, formatted in a manner very similar to that which the Scheduler returns.

Enabling the Quill daemon within Condor itself seems fairly straightforward, but there are several configuration settings in the configuration file which must be set before it will work properly. The PostgreSQL database itself was rather more difficult to set up, requiring the setup of several Quill-specific accounts and the creation of a database which would hold the information.

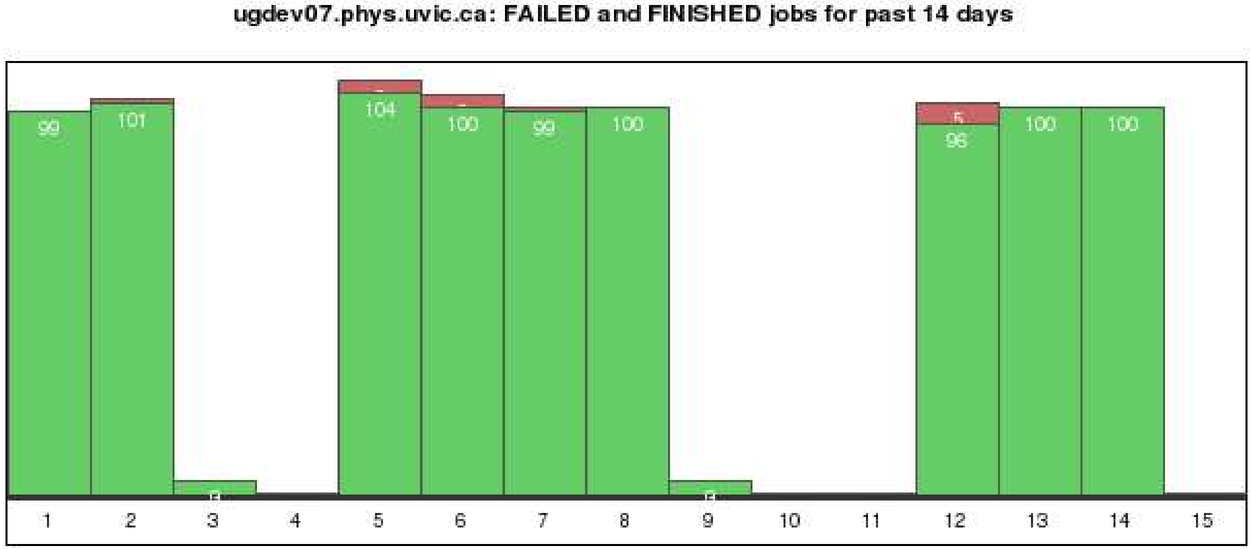


Figure 1: A graph of jobs run during testing, collected by the updated monitoring program. The green sections represent jobs successfully completed, while the red sections represent jobs which have failed.

The monitoring program which had been designed by my predecessor was complicated, and was applicable to only Condor History as output by the Scheduler daemon. In order to work with Quill, the parts of the program which interpreted the output given by the History command had to be altered. After incorporating the Quill commands, the monitoring script worked correctly, and testing began.

Testing of the grid was done by submitting approximately one hundred jobs per day to each head node at UVic, and recording the numbers of successful and failed jobs each day. This testing period also involved experimentation to determine what length of time would be most appropriate to leave between job submissions. By the end of this work term, failure rates had been reduced to approximately 1% on both of UVic's head nodes by pausing for sixty seconds between each submission. The altered job monitoring script took only five minutes to run what had previously been a three-hour procedure. An example plot from this testing period, showing success rates in green and failure rates in red, is given in Figure 2. The graph runs from the fifth of August (day 1) to the twentieth (day 15).

One disadvantage to the Quill database is that if, for any reason, the Quill is turned off or malfunctions, no jobs submitted during that time will appear on any Quill-based monitoring output. A job "missed" in this way can never be added to the output. This is generally considered an acceptable risk.

5. Personal Reflections

Working at the UVic Department of Physics and Astronomy was an enjoyable experience. My co-workers were very helpful, my hours were flexible, and working on the campus is very pleasant.

This job did not include as much programming as I expected, but the experience in troubleshooting was invaluable. I learned quite a bit about the Linux operating system and about networking, both of which should be very helpful in future jobs. More importantly than that, however, I learned first-hand, over many instances, how an interconnected computer system which is designed as well as it can be can sometimes not work despite the operators' best efforts. My experiences troubleshooting the grid have helped to develop my mindset as it relates to programming, and should serve me well in the future.

6. Conclusions

The Quill-based monitoring system for the GridX1 project is set up and operational and should need no further modifications. As in all systems, an upgrade will be required in time, but the nature of that upgrade is not in the scope of this report. The limitations of generating job monitor plots through queries to the Condor Scheduler have been alleviated through use of Condor Quill.

7. Acknowledgments

I would like to thank my supervisors, Ashok Agarwal and Randall Sobie; my predecessor, Tristan Sullivan; my co-workers, Daniel Vanderster and Ian Gable; the system administrator, Howard Peng; and my fellow co-op students, David Grundy and Cameron Sobie, for the help and support they have all given me these past four months.

References

Agarwal, Ashok, private communication, 2007.

Agarwal, Ashok, Ron Desmarais, Ian Gable, Sergey Popov, Sydney Schaffer, Cameron Sobie, Randall Sobie, Tristan Sulivan, Daniel Vanderster. BaBar MC Production on the Canadian Grid using a Web Services Approach. Computing in High Energy Physics (CHEP), 2007 in Victoria, Canada, submitted for presentation.

Agarwal et al. GridX1: A Canadian computational grid. Future Generation Computer Systems 23, issue 5, June 2007, Pages 680-687.

CANARIE. About CANARIE. http://www.canarie.ca/about/about.html , 2002.

Condor Project Team. What is Condor?. http://www.cs.wisc.edu/condor/description.html , 2007.

Popov, Sergey. HEP-Grid: The Missing Manual. https://wiki.gridx1.ca/twiki/bin/view/Main/MissingManual , 2006.

Portable Batch System Professional (2005). URL http://www.pbsgridworks.com/Default.aspx