Federating Grids: LCG Meets Canadian GridX1

R. Walker[1], M. Vetterli[1,2], R. Impey[3], G. Mateescu[3], B. Caron[2,4], A. Agarwal[5], A. Dimopoulos[5], L. Klektau[5], C. Lindsay[5], R.J. Sobie[5,6], D. Vanderster[5]

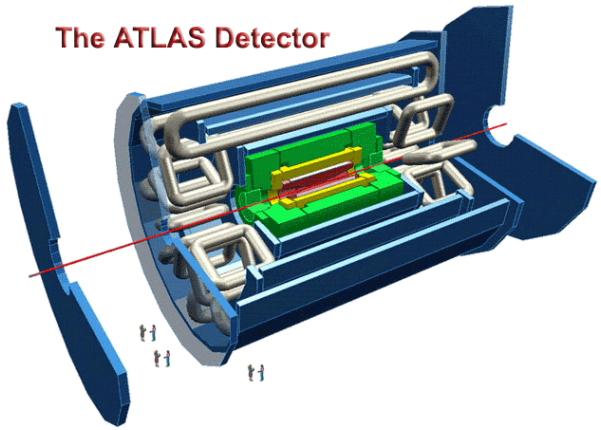
[1]Simon Fraser University, [2]TRIUMF, [3]National Research Council Canada, [4]University of Alberta, [5]University of Victoria, [6]Institute of Particle Physics of Canada

GlobusWorld 2005

Outline

Motivation

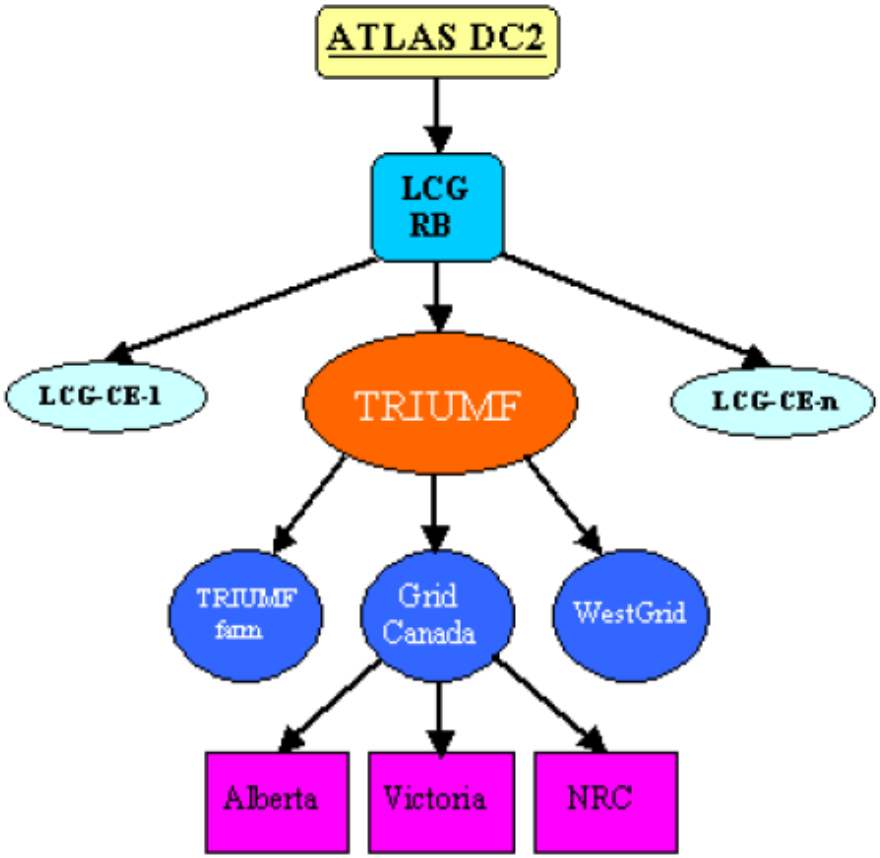
- LHC Compute Grid developed to analyse LHC experimental data in 2007.
- The LCG resources are dedicated to running LHC applications.
- ATLAS-Canada computing resources include:
  - Large processing and storage facility at TRIUMF.
  - A number of shared processing and storage facilities.



The Canadian Grid Model

- Each shared facility may have unique configuration requirements.
- Thus, we have the Canadian Grid model:
  - Generic Middleware (Virtual Data Toolkit: GT 2.4.3 + fixes)
  - No OS requirement: SuSe and RedHat clusters.
  - Generic user accounts: gcprod01 ... gcprod*mn*
  - Condor-G Resource Broker for load balancing.

System Overview



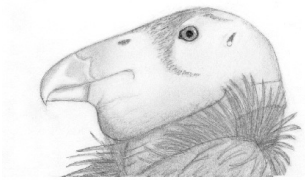
Outline

GridX1 Overview

- Currently have 3 clusters: UVic, UAlberta, and NRC.
- Over 200TB disk, 300 processors... more clusters soon.

Condor-G Overview

- Condor-G is an extension of the Condor batch system to the grid world.
- Provides intuitive commands to *submit*, *monitor*, and *cancel* tasks running on remote GRAM resources.
- Is known to be scalable to thousands of jobs.

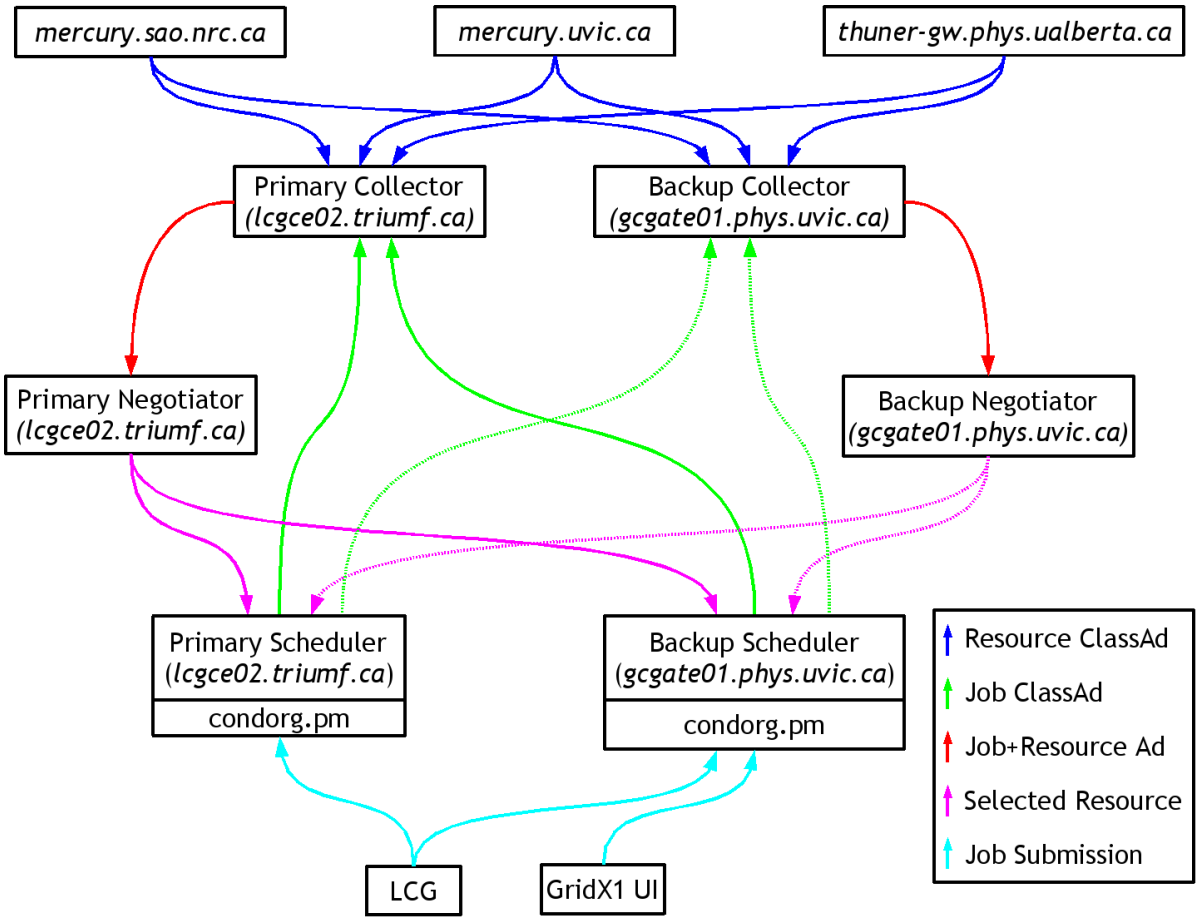


Condor
High Throughput Computing

Scheduling with Condor-G

1. Resources publish ClassAds to primary and backup *collectors*.
2. A user composes a job ClassAd.
3. The job is submitted to the primary or backup *scheduler*.
4. The job ClassAd is sent to the collector.
5. Collector passes all ClassAds to the *negotiator*.
6. Negotiator evaluates *Requirements* and *Rank* expressions.
7. The selected resource URL is returned to the scheduler.
8. The scheduler submits the job to the selected resource.

Condor-G Diagram



Resource ClassAds

- Each resource periodically publishes its state the primary and backup collectors.
- These ClassAds follow the GLUE Computing Element Schema:
  - Free and Total CPUs is published.
  - Number of running and waiting jobs is published.
- We also find and publish an estimated queue waiting time.
  - How long will a job wait if it is submitted to this resource?

Job ClassAds - Requirements

- Job ClassAds contain a resource *Requirements* expression.

- Prevent repetition of matched cluster:
  - *(TARGET.Name =!= LastMatchName0) ...*
- Processor availability:
  - *TARGET.gluecestatefreecpus>0*
- Enforce software versions:
  - *inList("VO-atlas-release-8.0.5"; TARGET.gluehostapplicationsoftwareruntimeenvironment; TARGET.Name) ==1.0*
- *inList* is an example of an *external* function.

Job ClassAds - Rank

- A *Rank* expression is used to evaluate the relative utility of each resource.
- Typical *Rank* is the negative EWT - high waiting times give a low rank.
  - *Rank = 0.-TARGET.gluecestateestimatedresponsetime - (TARGET.CurMatches * 60.)*
  - (assume 60 second penalty for current matches)
- We can incorporate data location using *external* function:
  - *Rank = 0.-TARGET.gluecestateestimatedresponsetime - (TARGET.CurMatches * 60.) - ((1. - dataOverlap(TARGET.Name;"dc2.003103.evgen....root")) * 200.)*

Outline

The Globus JobManager: An Interface

- The Globus JobManager provides an interface to a resource.
- Every GRAM gatekeeper has a JobManager.
- A JobManager is written specifically to the local resource management system (e.g. PBS, Condor, CondorG, etc...).
- Implements *submit*, *status*, *remove* commands to execute the job.

- To make GridX1 a single resource to LCG, we use CondorG as our LRMS.

The LCG/GridX1 Interface

- The GridX1 Interface is a standard LCG Compute Element with a CondorG JobManager and Information Provider.
- The CondorG JobManager does the following:
  - creates the CondorG job description file
  - submits to the Condor scheduler using `condor_submit`
  - polls the job using `condor_q`
  - retrieves a full proxy from a MyProxy server.

Authorisation for 2nd GRAM Submission

- Having a 2nd GRAM submission creates a proxy issue.
- GRAM submission from the LCG RB delegates a *limited* proxy.
  - This proxy can be used for GridFTP, but not a further GRAM submission.
- We need to acquire a *full* proxy for the 2nd submission.

- We could delegate a full proxy via GRAM, but we have chosen a different solution.
- For the ATLAS application: users must store her credentials in a known MyProxy server.
- The limited proxy is used to delegate a full proxy via MyProxy.

LCG Requirements for Worker Nodes

- In general, we can repackage jobs for submission to the sub-Grid.
  - i.e. produce a wrapper script, package the software, stage in data.
- For ATLAS we pass the job *as-is*, thus we have some worker node requirements:
  - DC2 application must be installed.
  - LCG data-handling tools must be installed: edg-rm, lcg-xx.
  - Outbound network connectivity for the GT2 client for data I/O (sandboxes, storage element data, conditions DB).

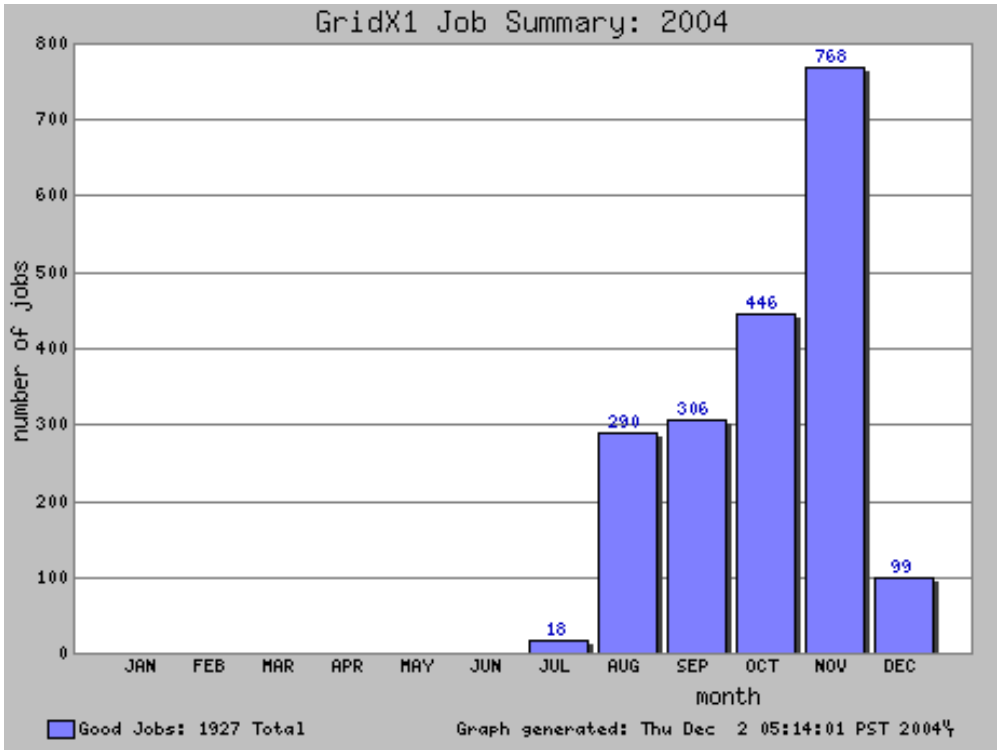
Outline

Application Performance and Reliability

- ATLAS DC2 has been run effectively on 1300 processors using this interface.
- Success/Failure ratio similar to overall LCG.
- GC Failures are well understood: proxy issues, firewalls, etc...
- Faulty clusters can be easily avoided.

Application Performance and Reliability

- Over 2000 jobs have been run successfully using this system.



Application Monitoring

Grid X1 Condor-G Monitoring

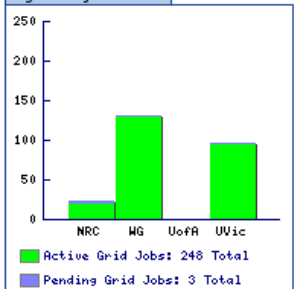
Resource Status

resource	pending jobs	active jobs	est. wait time	last updated
hep.westgrid.ca	0	129	00:00:00	2 Dec, 10:10
mercury.sao.nrc.ca	2	21	22:35:28	2 Dec, 10:10
mercury.uvic.ca	0	97	00:00:00	2 Dec, 10:10
thuner-gw.phys.ualberta.ca	0	0	00:00:00	2 Dec, 10:05
grid totals	2	247	00:00:00	

Job Status

job id	user id	owner	command	resource	status	run time	time submitted
7774.0	atlas006	frederic brochu	data 'UI= ...'	mercury.uvic.ca	ACTIVE	44:22:21	30 Nov, 14:44
7902.0	atlas006	frederic brochu	data 'UI= ...'	mercury.sao.nrc.ca	ACTIVE	27:55:21	1 Dec, 07:18
7911.0	atlas006	frederic brochu	data 'UI= ...'	mercury.uvic.ca	ACTIVE	17:51:22	1 Dec, 07:20
7926.0	atlas006	frederic brochu	data 'UI= ...'	hep.westgrid.ca	ACTIVE	24:32:21	1 Dec, 10:27
7949.0	atlas006	frederic brochu	data 'UI= ...'	mercury.uvic.ca	ACTIVE	24:16:30	1 Dec, 10:56
7956.0	atlas006	frederic brochu	data 'UI= ...'	hep.westgrid.ca	ACTIVE	21:35:20	1 Dec, 13:29
7972.0	atlas006	frederic brochu	data 'UI= ...'	mercury.uvic.ca	ACTIVE	18:17:47	1 Dec, 16:50
7973.0	atlas006	frederic brochu	data 'UI= ...'	hep.westgrid.ca	ACTIVE	18:18:18	1 Dec, 16:50
7974.0	atlas006	frederic brochu	data 'UI= ...'	hep.westgrid.ca	ACTIVE	12:08:55	1 Dec, 16:51
7975.0	atlas006	frederic brochu	data 'UI= ...'	mercury.uvic.ca	ACTIVE	18:17:52	1 Dec, 16:53
7977.0	atlas006	frederic brochu	data 'UI= ...'	mercury.sao.nrc.ca	ACTIVE	18:19:42	1 Dec, 16:54
7983.0	atlas006	frederic brochu	data 'UI= ...'	hep.westgrid.ca	ACTIVE	18:09:21	1 Dec, 16:56
7987.0	atlas003	Rodney Walker	data 'UI= ...'	Not Matched	UNSUBMITTED		1 Dec, 17:04
7988.0	atlas006	frederic brochu	data 'UI= ...'	mercury.uvic.ca	ACTIVE	17:42:22	1 Dec, 17:26
7989.0	atlas006	frederic brochu	data 'UI= ...'	hep.westgrid.ca	ACTIVE	17:39:20	1 Dec, 17:26
7991.0	atlas006	frederic brochu	data 'UI= ...'	hep.westgrid.ca	ACTIVE	17:34:20	1 Dec, 17:27
7996.0	atlas006	frederic brochu	data 'UI= ...'	mercury.uvic.ca	ACTIVE	16:26:46	1 Dec, 18:40
7997.0	atlas006	frederic brochu	data 'UI= ...'	hep.westgrid.ca	ACTIVE	16:31:45	1 Dec, 18:40

gridx1 job load



grid monitor

thuner-gw.phys.ualberta.ca
Dec 2, 2004 10:01:13 PST
Job Submission Up
Gatekeeper Up
GridFTP Up

mercury.uvic.ca
Dec 2, 2004 10:01:06 PST
Job Submission Up
Gatekeeper Up
GridFTP Up

mercury.sao.nrc.ca
Dec 2, 2004 10:02:00 PST
Job Submission Up
Gatekeeper Up
GridFTP Up

Conclusions

- Demonstrated a working federated Grid.
- Made available 1300 CPUs at 4 shared facilities to ATLAS DC2.
- Allows for fast and easy deployment of new resources.

Future Work: Data Handling Tools

- Data handling tools will help us manage our storage facilities.
- Using RLS tools, we can incorporate data-locations to the RB process.
  - *Requirements* and *Rank* expressions modified to use data metric.
- We have developed an *external-Rank* function to query the RLS for data availability.

Future Work: BaBar SPGrid

- Canada has produced approx. 20% of BaBar SP6 events.
- BaBar SP6 production clusters: UVic Muse, UVic Mercury, WestGrid (UBC).

- SP production will be grid-enabled using the system described here.
- Each cluster will require a BaBar machine for conditions DB.
- Jobs could be submitted from BaBar to the GridX1 interface.