Grid Canada Testbed using HEP applications

Randall Sobie

A.Agarwal, J.Allan, M.Benning, G.Hicks, R.Impey, R.Kowalewski, G.Mateescu, D.Quesnel, G.Smecher, D.Vanderster, I.Zwiers

Institute for Particle Physics, University of Victoria
National Research Council of Canada, CANARIE
BC Ministry for Management Services

Outline

Introduction
Grid Canada Testbed
HEP Applications
Results
Conclusions

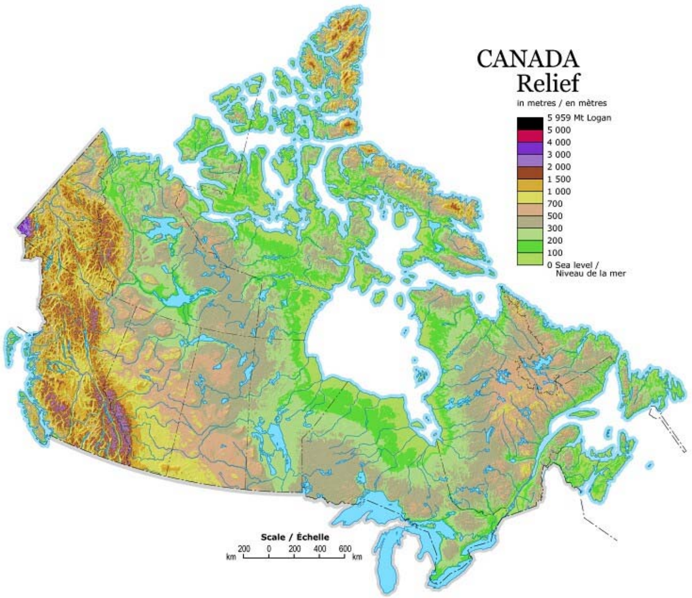
Introduction

Learn to establish and maintain an operating Grid in Canada

Learn how to run our particle physics apps on the Grid

- **BaBar simulation**
- **ATLAS data challenge simulation**

Significant computational resources being installed on condition that they share 20% of their resources



Exploit the computational resources available at both HEP and non-HEP sites without installing application-specific software at each site

Grid Canada

Grid Canada was established to foster Grid research in Canada

Sponsored by CANARIE, C3.Ca Association and National Research Council of Canada

Activities:

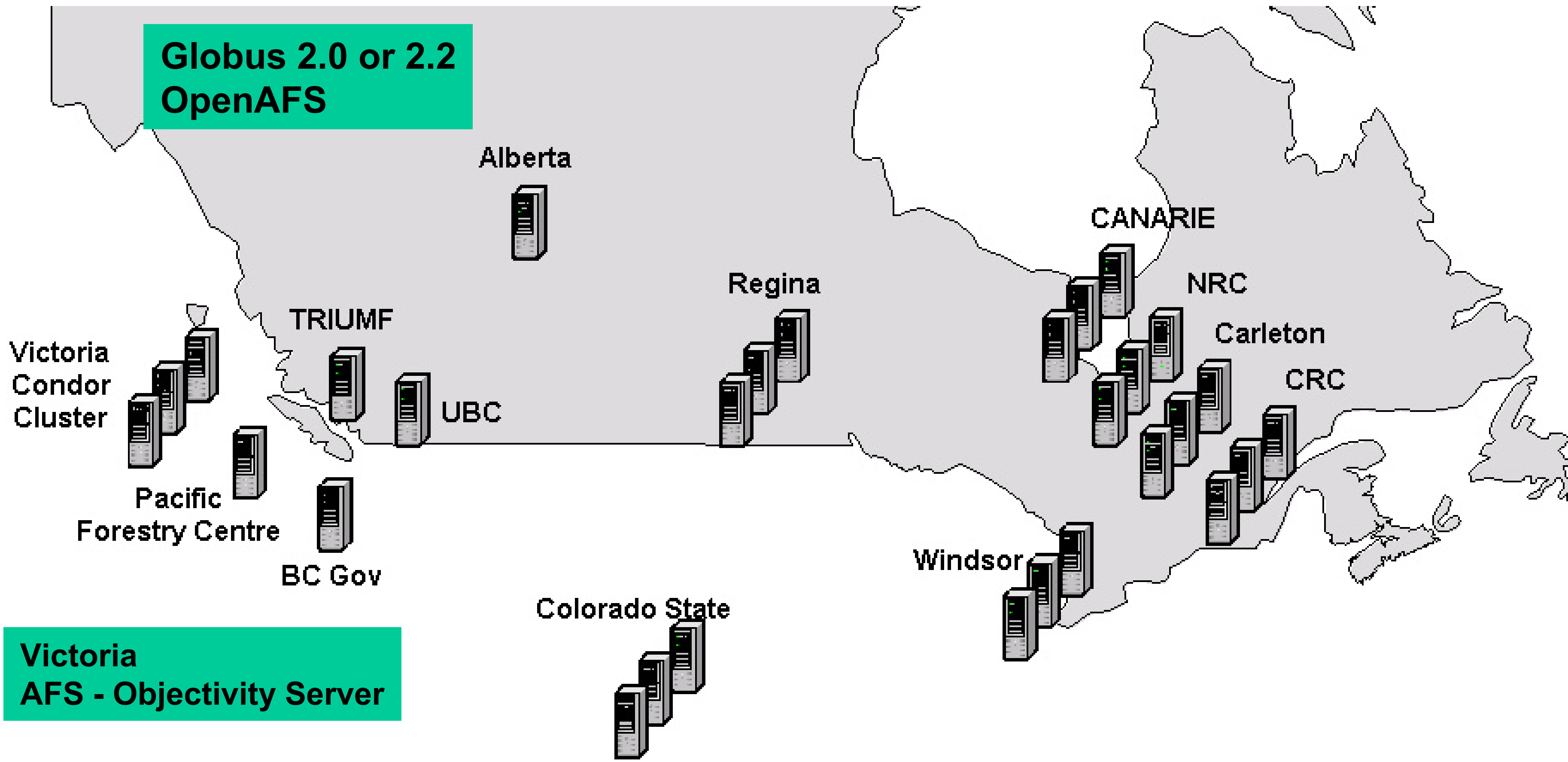
1. Operates the Canadian Certificate Authority
2. HPC Grid testbed for parallel applications
3. Linux Grid testbed
4. High speed network projects
   - TRIUMF-CERN 1 TB file transfer demo (iGrid)

Grid Canada Linux Testbed

12 sites across Canada (+ 1 in Colorado)

1-8 nodes per site (mixture of single and clusters of machines)

Network connectivity 10-100 Mbps from each site to Victoria Servers



HEP Simulation Applications

Event Generator

↓

Detector simulation

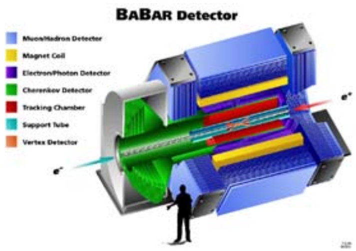
↓ **Inject Background evts**

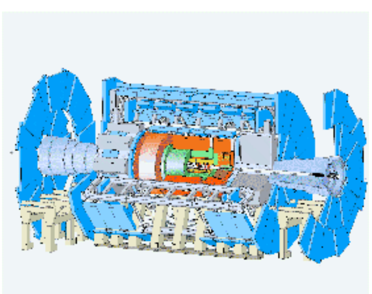
Reconstruction

Simulation of event data is done similarly between all HEP experiments.

Each step is generally a separate job.

BaBar uses an Objectivity DB for the event store





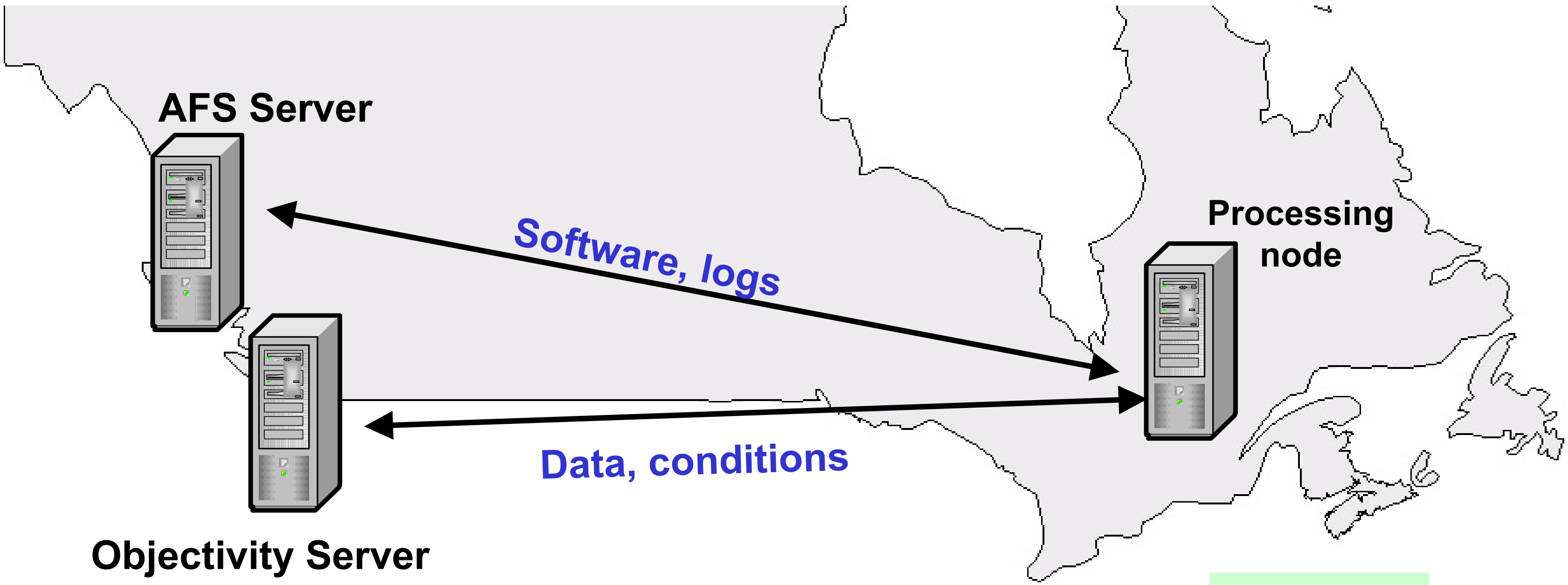
ATLAS data challenge (DC1) uses Zebra

Neither application are optimized for a Wide-Area Grid

Objectivity DB Application

3 parts to the job
(event generation, detector simulation and reconstruction)
4 hrs for 500 events on a 450 MHz CPU

1-day tests consisted of 90-100 jobs (~50,000 evts) using 1000 SI95

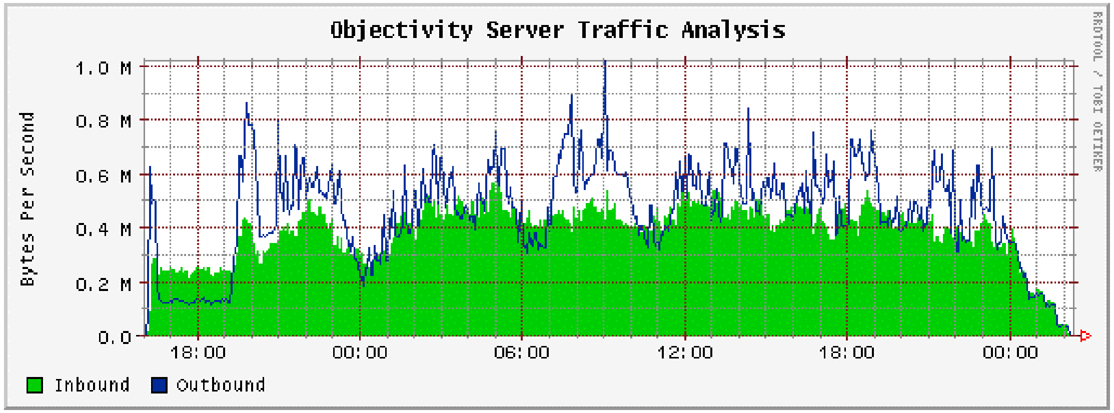


100 Objy contacts per event

Latencies ~100ms

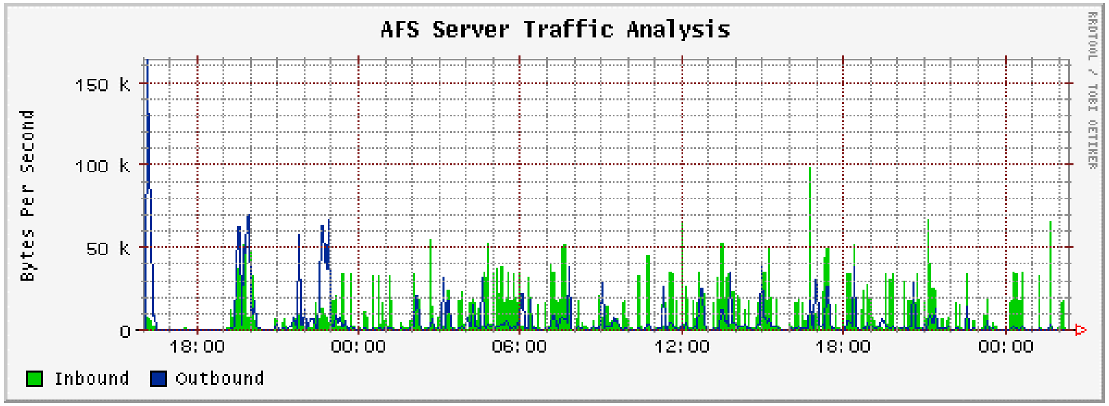
Results

A series of 1-day tests of the entire testbed using 8-10 sites
80-90% success rate for jobs



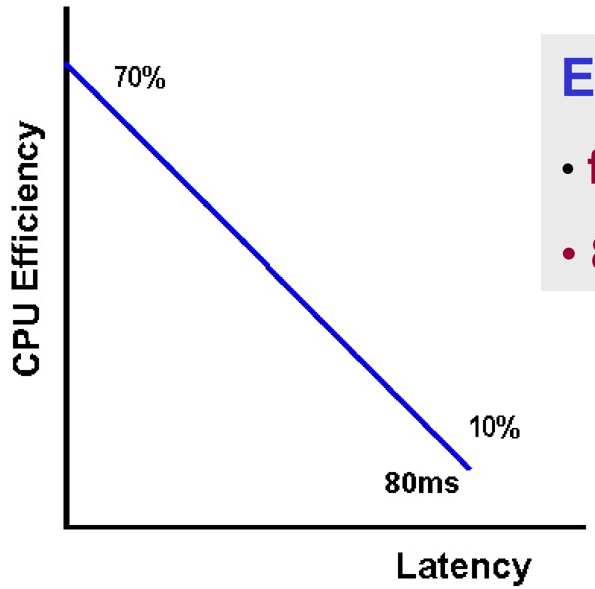
Objectivity

- DB lock problems - container creation requires global lock
- “lock-monitor”
- “lock-cleaner” at end of job



AFS – worked well

- network demand high during startup (some crashes due to inability to access AFS)



Efficiency was low at distant sites

- frequent DB access for reading/writing data
- 80ms latencies

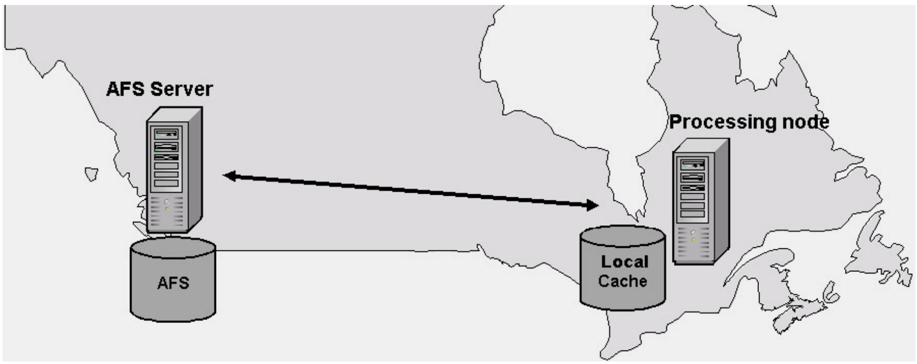
Next step?

- fix application so it has less frequent DB access
- install multiple Objectivity servers at different sites

HEP appears to be moving away from Objy

Typical HEP Application

Input events and output are read/ written into standard files (eg Zebra, Root)



Software is accessed via AFS from Victoria server.

No application dependent software at hosts.

We explored 3 operating scenarios:

1. **AFS for reading and writing data**
2. **GridFTP input data to site then write output via AFS**
3. **GridFTP both input and output data**

AFS for reading and writing data

AFS is the easiest way to run the application over the grid however its performance was poor as noted by many groups.

In particular, frequent reading of input data via AFS was poor
Remote CPU utilization < 5%

GridFTP input data to site and write output via AFS

AFS caches its output on local disk and then transfers to server.

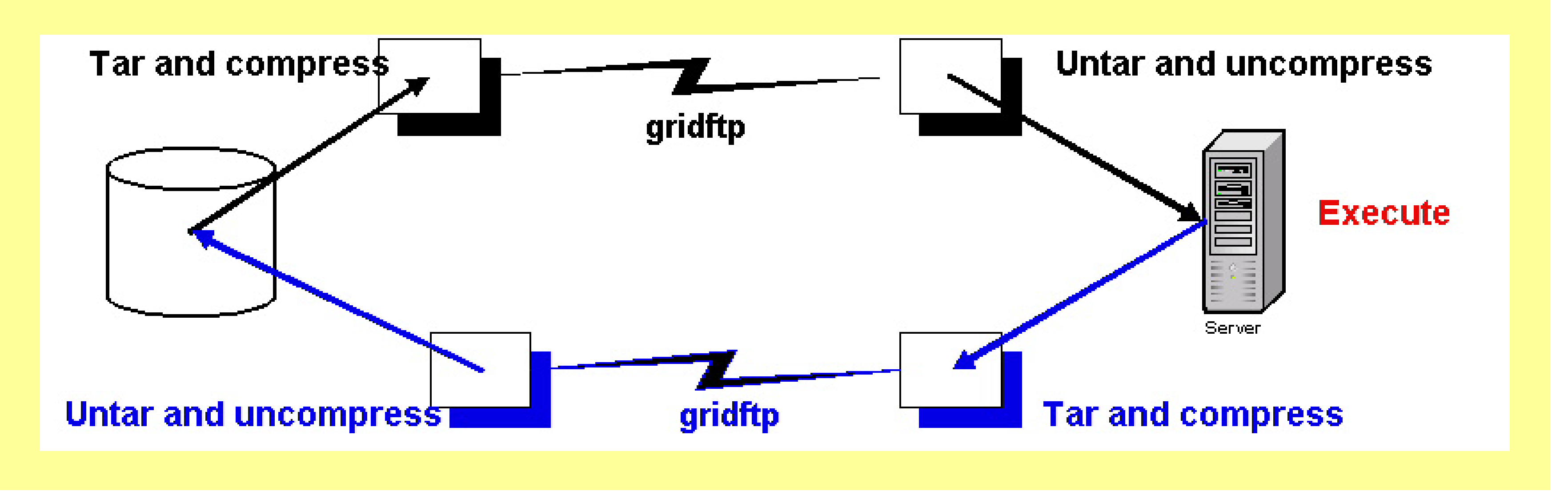
AFS transfer speeds were close to single-stream FTP

Neither were considered to be optimal for production over the Grid

GridFTP both input and output data (Software via AFS)

- AFS used to access static executable (400 MB) and for log files

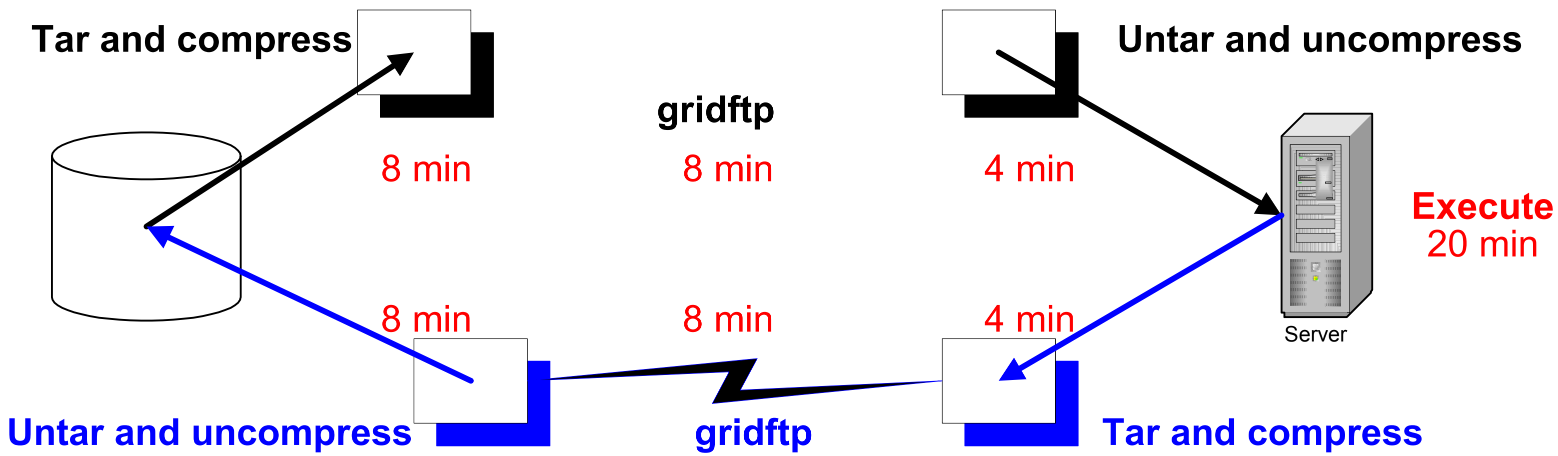
- GridFTP for tarred and compressed input and output files
  - input 2.7 GB (1.2 GB compressed)
  - output 2.1 GB (0.8 GB compressed)



Results

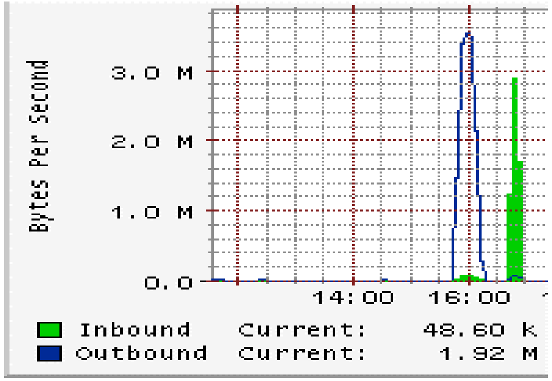
Currently we have run this application over a subset of the Grid Canada testbed with machines local, 1500km and 3000km.

We use a single application executes quickly. (ideal for grid tests)



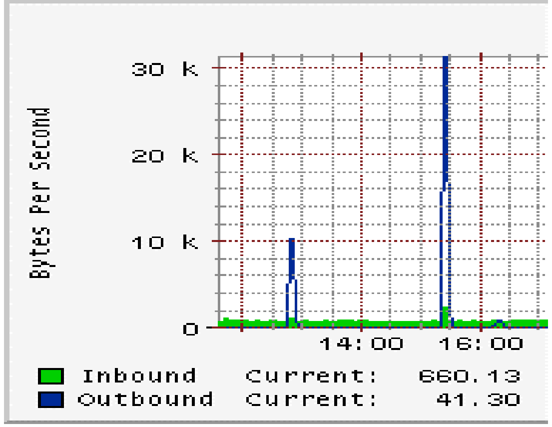
Typical times for running the application at a 3000km distant site.

Network and local cpu utilization.



Network traffic on the GridFTP machine for a single application

Typical transfer rates ~ 30 mbits/s



Network traffic on the AFS Server

Little demand on AFS

Plan is to run multiple jobs at all sites on GC Testbed

Jobs are staggered to reduce initial I/O demand

- **Normally jobs would read different input files**

We do not see any degradation in CPU utilization due to AFS.

- **It may become an issue with more machines - we are running 2 AFS servers.**
- **We could improve AFS utilization by running an mirrored remote site**

We may become network-limited as the number of applications increase.

Success ? ***This is a mode of operation that could work***

It appears that the CPU efficiency at remote sites is 80-100% (not limited by AFS)

Transfer rate of data is (obviously) limited by the network capacity.

We can run our HEP applications without any more than Linux, Globus and AFS-Client.

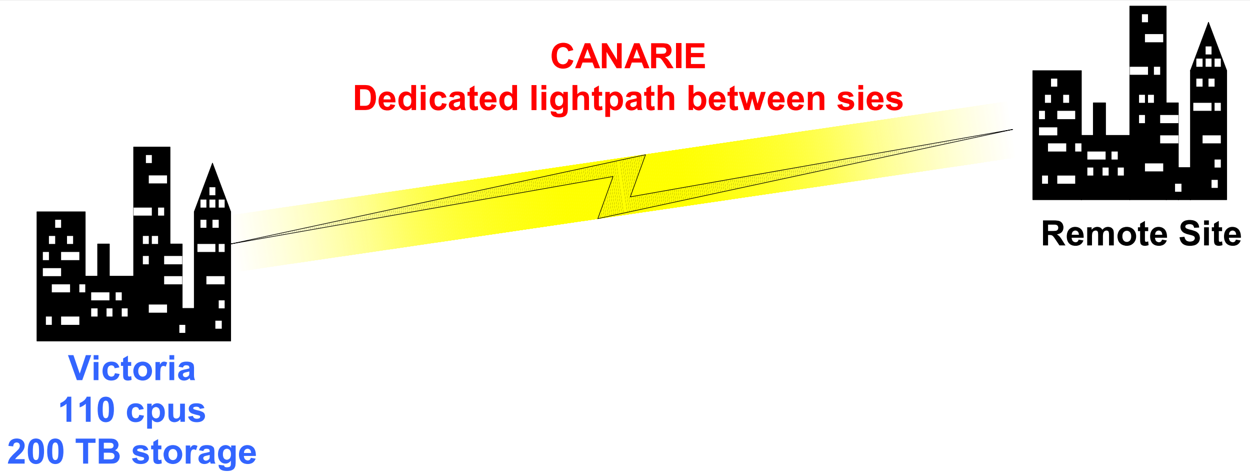
Next Steps

We have been installing large, new computational and storage facilities both shared and dedicated to HEP as well as a new high speed network.

We believe we understand the basic issues in running a Grid but there is lots to do

- **we do not run a resource broker**
- **error and fault detection is minimal or non-existent**
- **our applications could be better tuned to run over the Grid testbed**

The next step will likely involve fewer sites, but more CPUs with the goal of making a more production-type facility.



Summary

- **Grid Canada testbed has been used to run HEP applications at non-HEP sites**
  - **Require only Globus, AFS-Client at remote Linux CPU**
  - **Input/Output data transferred via GridFTP**
  - **Software accessed by AFS**

- **Continuing to test our applications at a large number of widely distributed sites**
  - **Scaling issues so far have not been a problem but we are still using relatively few resources (10-20 CPUs)**

- **Plan to utilize new computational and storage resources with the new CANARIE network to develop a production Grid**

Thanks to the many people who have established and worked on the GC testbed and/or provided access to their resources.