Operating a distributed IaaS Cloud

Ian Gable

Ashok Agarwal, Patrick Armstrong Adam Bishop, Andre Charbonneau, Ronald Desmarais, Kyle Fransham, Roger Impey, Colin Leavett-Brown, Michael Paterson, Wayne Podaima, Randall Sobie, Matt Vliet

University of Victoria, Victoria, Canada

National Research Council of Canada, Ottawa

HEPiX Spring 2011, GSI

Outline

- Motivation
  - – HEP Legacy Data Project
  - – CANFAR: Observational Astronomy
- System Architecture
- Operational Experience
- Future work
- Summary

Motivation

- Projects requiring modest resources we believe to be suitable to Infrastructure-as-a-Service (IaaS) Clouds:
  - The High Energy Physics Legacy Data project
  - The Canadian Advanced Network for Astronomical Research (CANFAR)
- We expect an increasing number of IaaS clouds to be available for research computing.

HEP Legacy Data Project

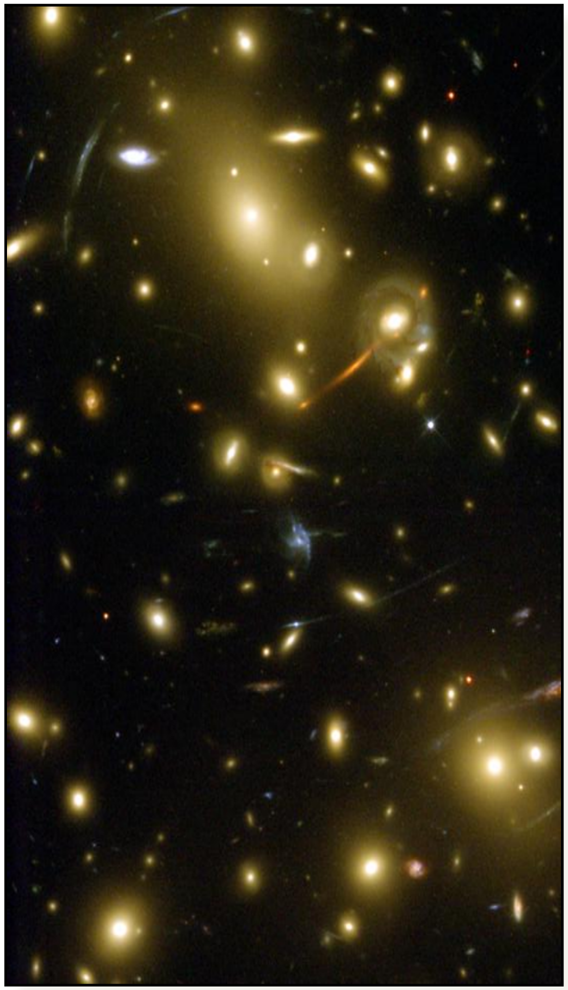
- We have been funded in Canada to investigate a possible solution for analyzing BaBar data for the next 5-10 years.
- Collaborating with SLAC who are also pursuing this goal.
- We are exploiting VMs and IaaS clouds.
- Assume we are going to be able run BaBar code in a VM for the next 5-10 years.
- We hope that results will be applicable to other experiments.
- 2.5 FTEs for 2 years ends in October 2011.

- 9.5 million lines of C++ and Fortran
- Compiled size is 30 GB
- Significant amount of manpower is required to maintain the software
- Each installation must be validated before generated results will be accepted
- Moving between SL 4 and SL 5 required a significant amount of work, and is likely the last version of SL that will be supported

CANFAR

Canadian Advanced Network for Astronomical Research

- CANFAR is a partnership between
  - University of Victoria
  - University of British Columbia
  - National Research Council Canadian Astronomy Data Centre
  - Herzberg Institute for Astrophysics
- Will provide computing infrastructure for 6 observational astronomy survey projects



CANFAR

Canadian Advanced Network for Astronomical Research

- Jobs are embarrassingly parallel, much like HEP.
- Each of these surveys requires a different processing environment, which require:
  - A specific version of a Linux distribution
  - A specific compiler version
  - Specific libraries
- Applications have little documentation
- These environments are evolving rapidly

How do we manage jobs on IaaS?

- With IaaS, we can easily create many instances of a VM image
- How do we Manage the VMs once booted?
- How do we get jobs to the VMs?

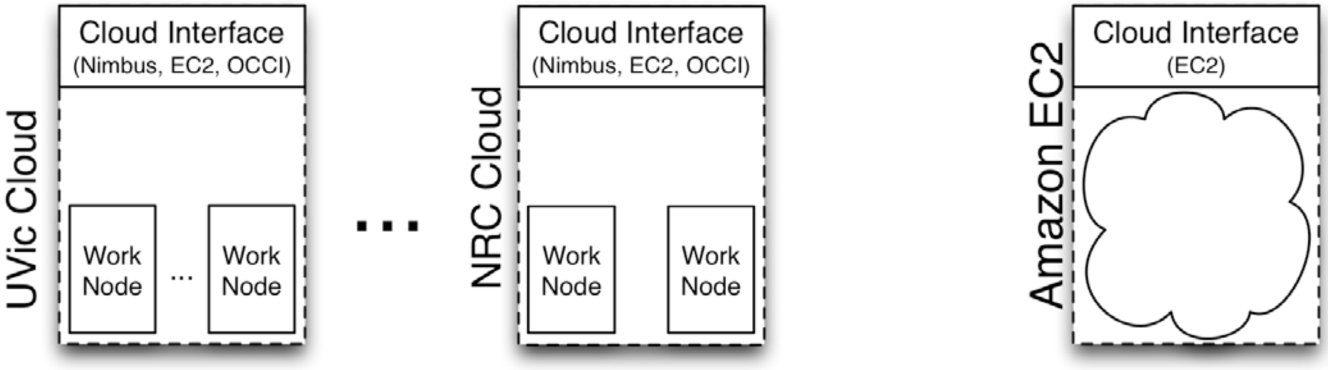
Possible solutions

- The Nimbus Context broker allows users to create “One Click Clusters”
  - Users create a cluster with their VM, run their jobs, then shut it down
  - However, most users are used to sending jobs to a HTC cluster, then waiting for those jobs to complete
  - Cluster management is unfamiliar to them
  - Already used for a big run with STAR in 2009
- Univa Grid Engine Submission to Amazon EC2
  - Release 6.2 Update 5 can work with EC2
  - Only supports Amazon
- This area is involving very rapidly!
- Other solutions?

Our Solution: Condor + Cloud Scheduler

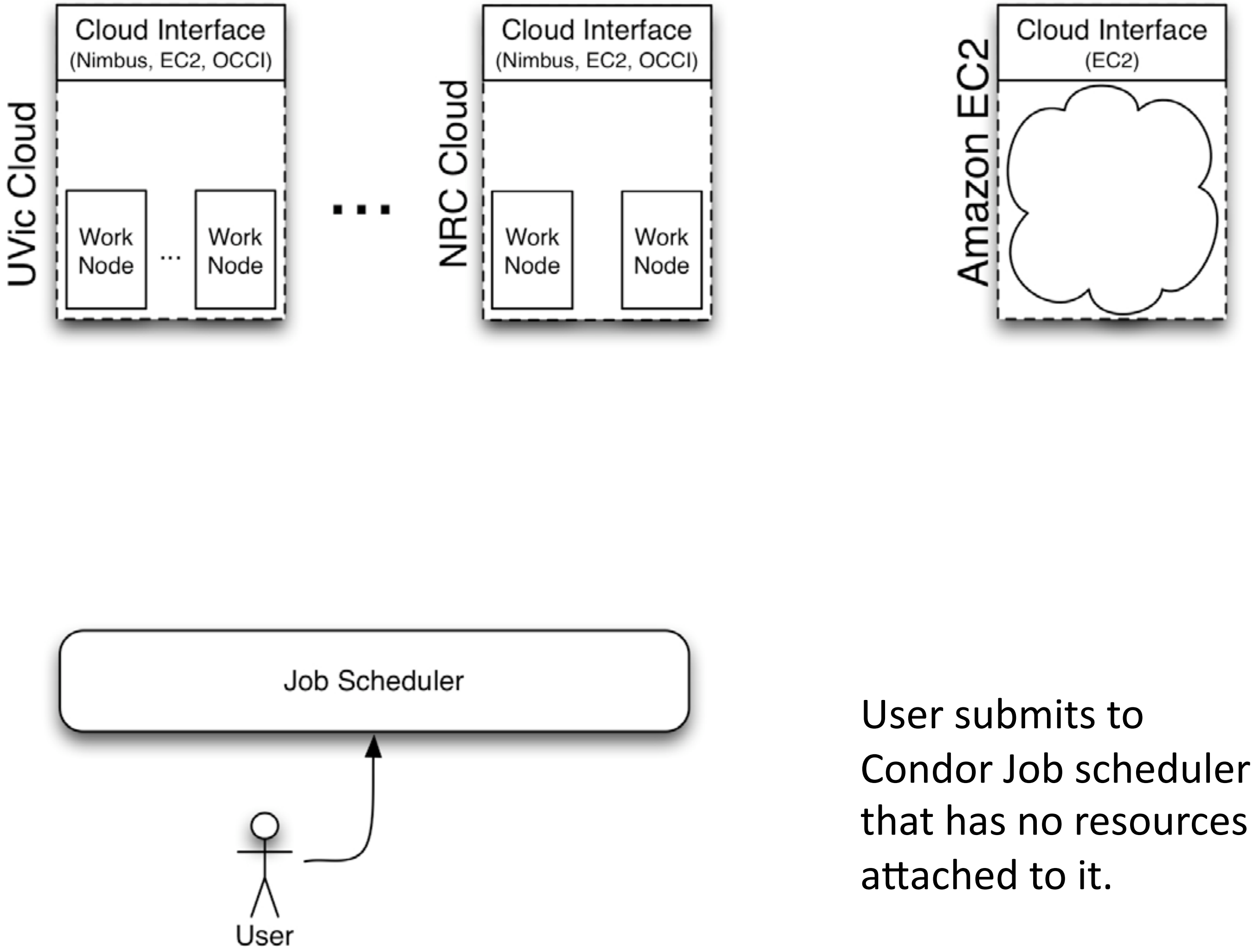
- Users create a VM with their experiment software installed
  - A basic VM is created by our group, and users add on their analysis or processing software to create their custom VM
- Users then create batch jobs as they would on a regular cluster, but they specify which VM should run their images
- Aside from the VM creation step, this is very similar to the HTC workflow

Step 1



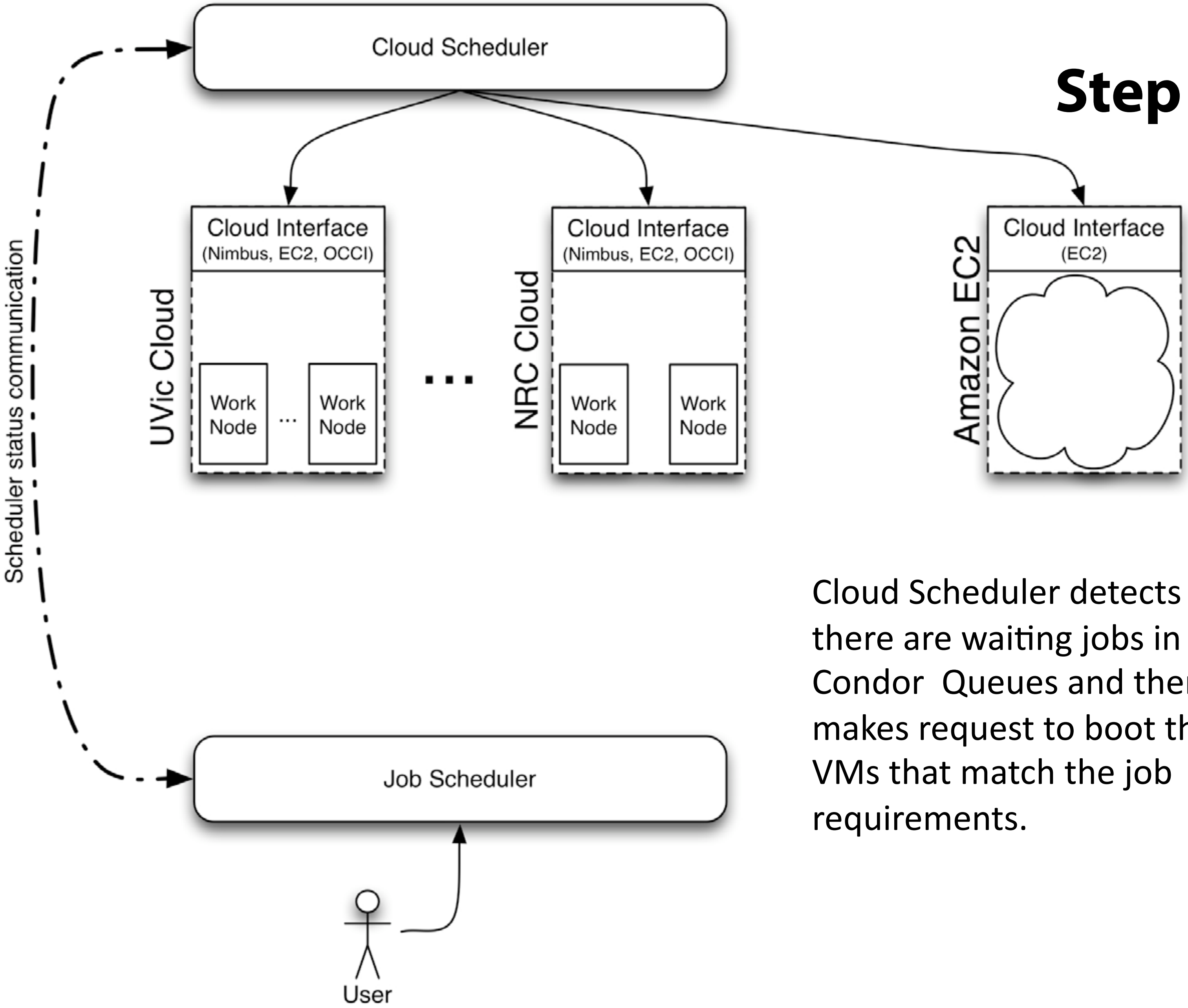
Research and Commercial clouds made available with some cloud-like interface.

Step 2



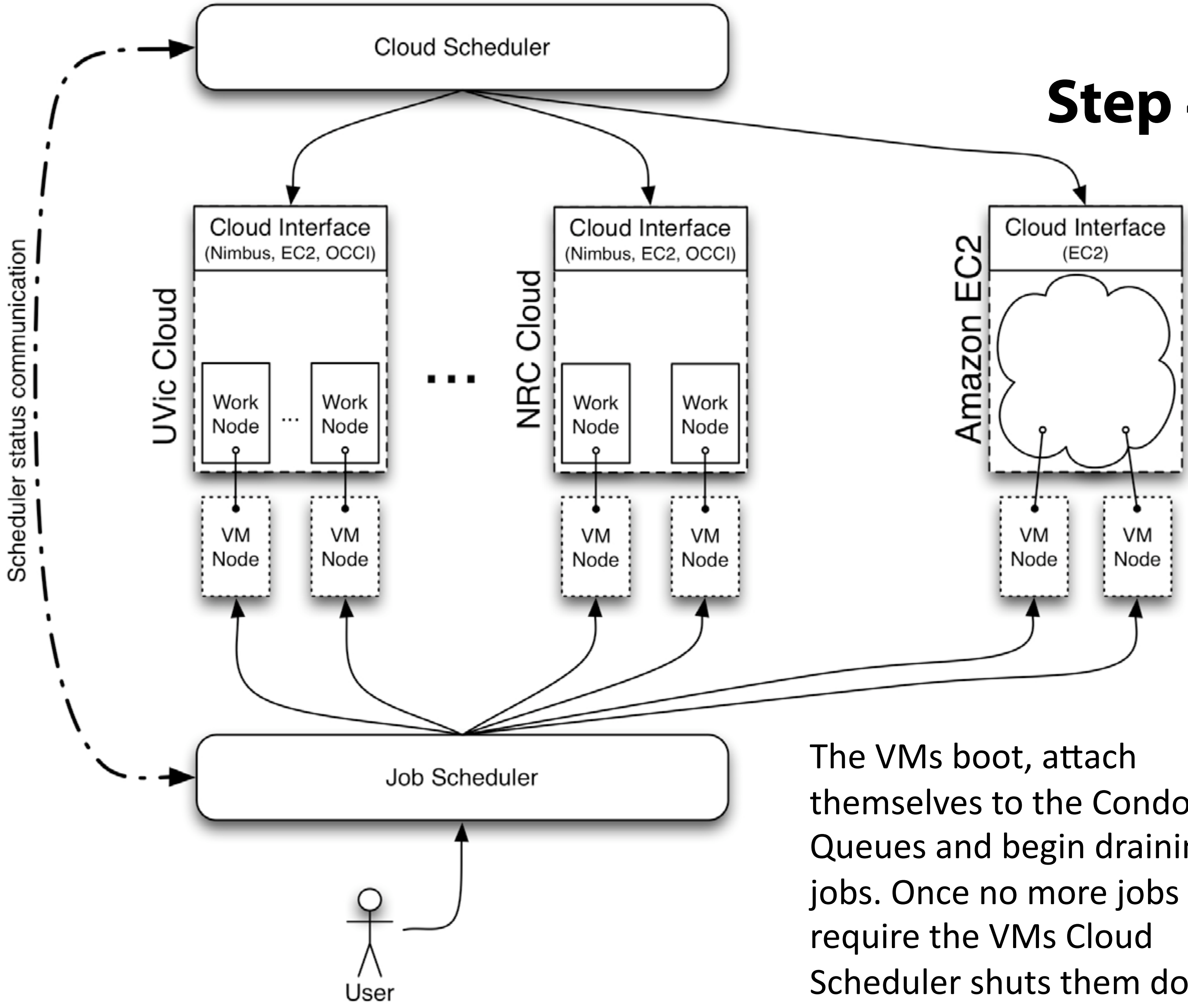
User submits to Condor Job scheduler that has no resources attached to it.

Step 3



Cloud Scheduler detects that there are waiting jobs in the Condor Queues and then makes request to boot the VMs that match the job requirements.

Step 4



The VMs boot, attach themselves to the Condor Queues and begin draining jobs. Once no more jobs require the VMs Cloud Scheduler shuts them down.

How does it work?

1. A user submits a job to a job scheduler
2. This job sits idle in the queue, because there are no resources yet
3. Cloud Scheduler examines the queue, and determines that there are jobs without resources
4. Cloud Scheduler starts VMs on IaaS clusters
5. These VMs advertise themselves to the job scheduler
6. The job scheduler sees these VMs, and starts running jobs on them
7. Once all of the jobs are done, Cloud Scheduler shuts down the VMs

Implementation Details

- We use Condor as our job scheduler
  - Good at handling heterogeneous and dynamic resources
  - We were already familiar with it
  - Already known to be scalable
- We use Condor Connection broker to get around private IP clouds
- Primarily support Nimbus and Amazon EC2, with experimental support for OpenNebula and Eucalyptus.

Implementation Details Cont.

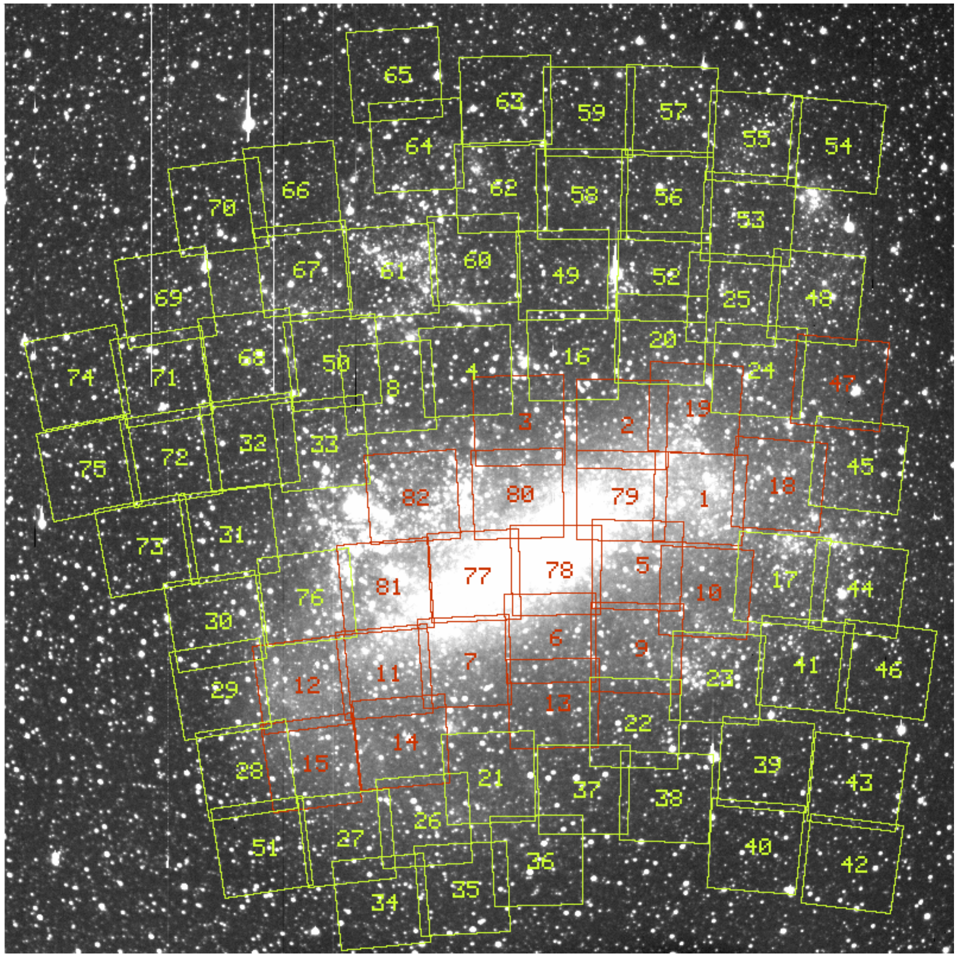
- Each VM has the Condor startd daemon installed, which advertises to the central manager at start
- We use a Condor Rank expression to ensure that jobs only end up on the VMs they are intended to
- Users use Condor attributes to specify the number of CPUs, memory, scratch space, that should be on their VMs
- We have a rudimentary round robin fairness scheme to ensure that users receive a roughly equal share of resources respects condor priorities

Condor Job Description File

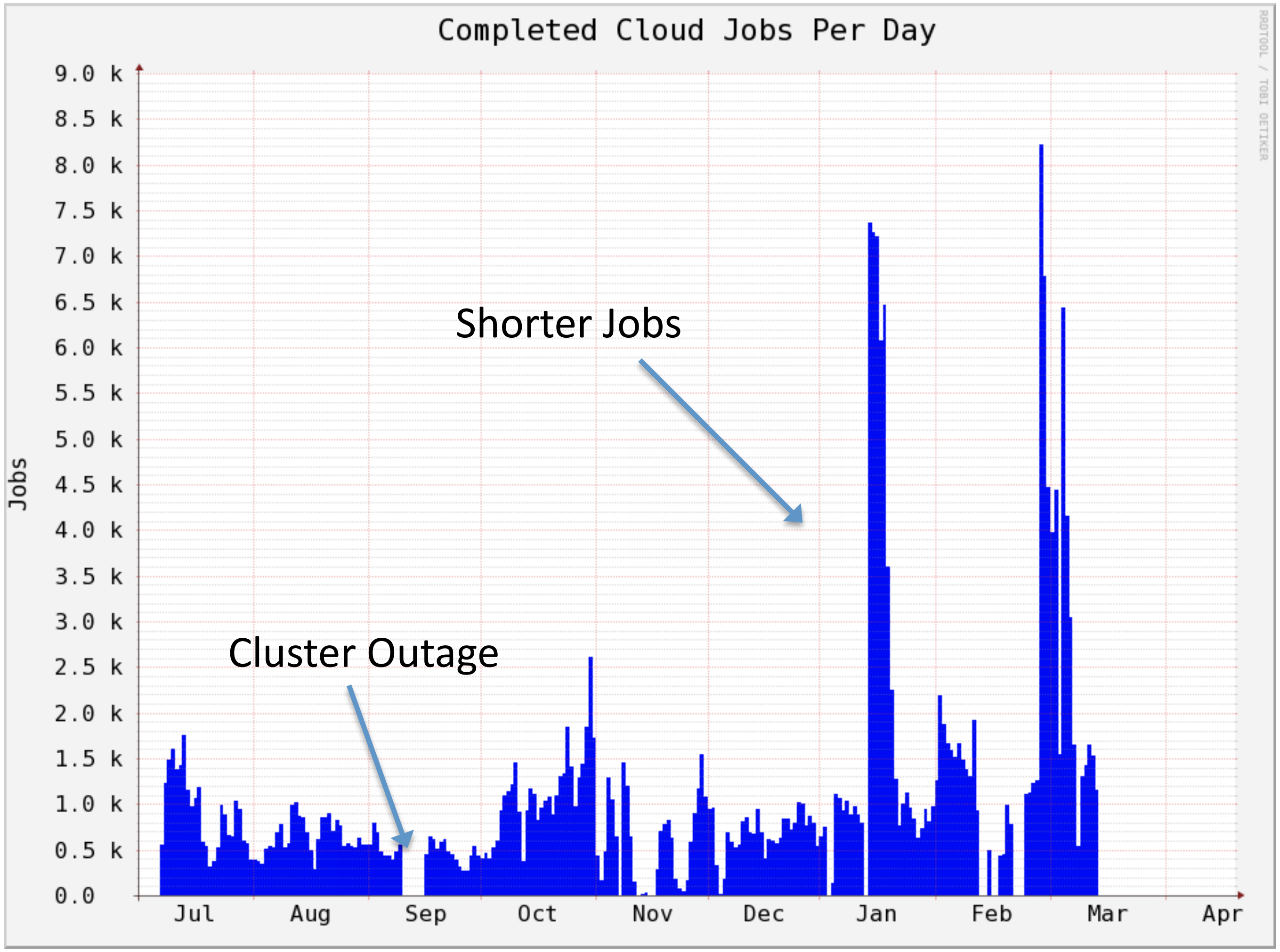
```
Universe = vanilla
Executable = red.sh
Arguments = W3-3+3 W3%2D3%2B3
Log = red10.log
Output = red10.out
Error = red10.error
should_transfer_files = YES
when_to_transfer_output = ON_EXIT

# Run-environment requirements
Requirements = VMType =?= ”redshift"
+VMNetwork = "private"
+VMCPUArch = "x86"
+VMLoc = "http://vmrepo.phys.uvic.ca/vms/
redshift.img.gz"
+VMMem = ”2048"
+VMCPUCores = "1"
+VMStorage = "20"
+VMAMI = "ami-fdee0094"
Queue
```

CANFAR: MAssive Compact Halo Objects

- Detailed re-analysis of data from the MACHO experiment Dark Matter search.
- Jobs perform a wget to retrieve the input data (40 M) and have a 4-6 hour run time. Low I/O great for clouds.
- Astronomers happy with the environment.



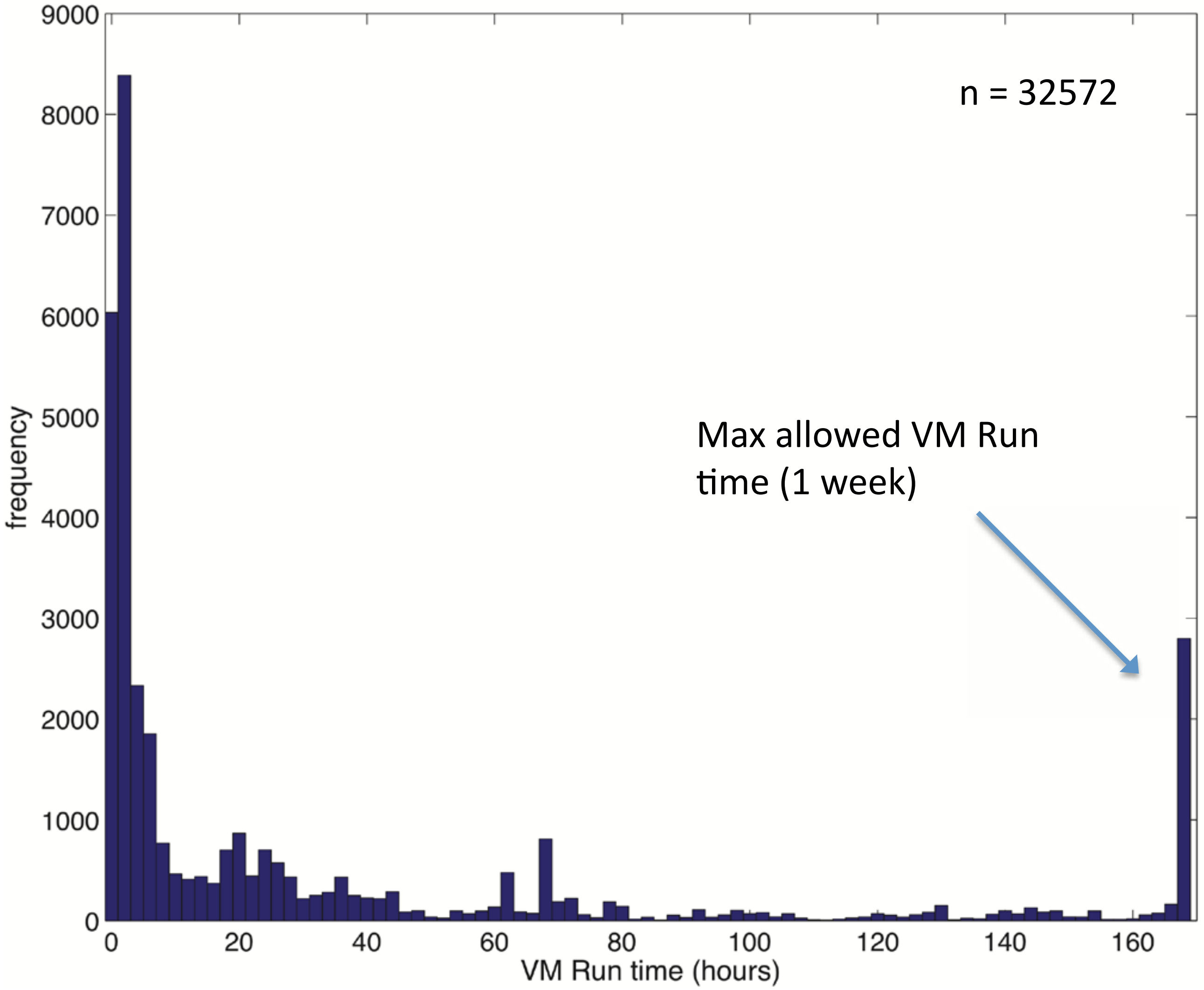
Experience with CANFAR



Shorter Jobs

Cluster Outage

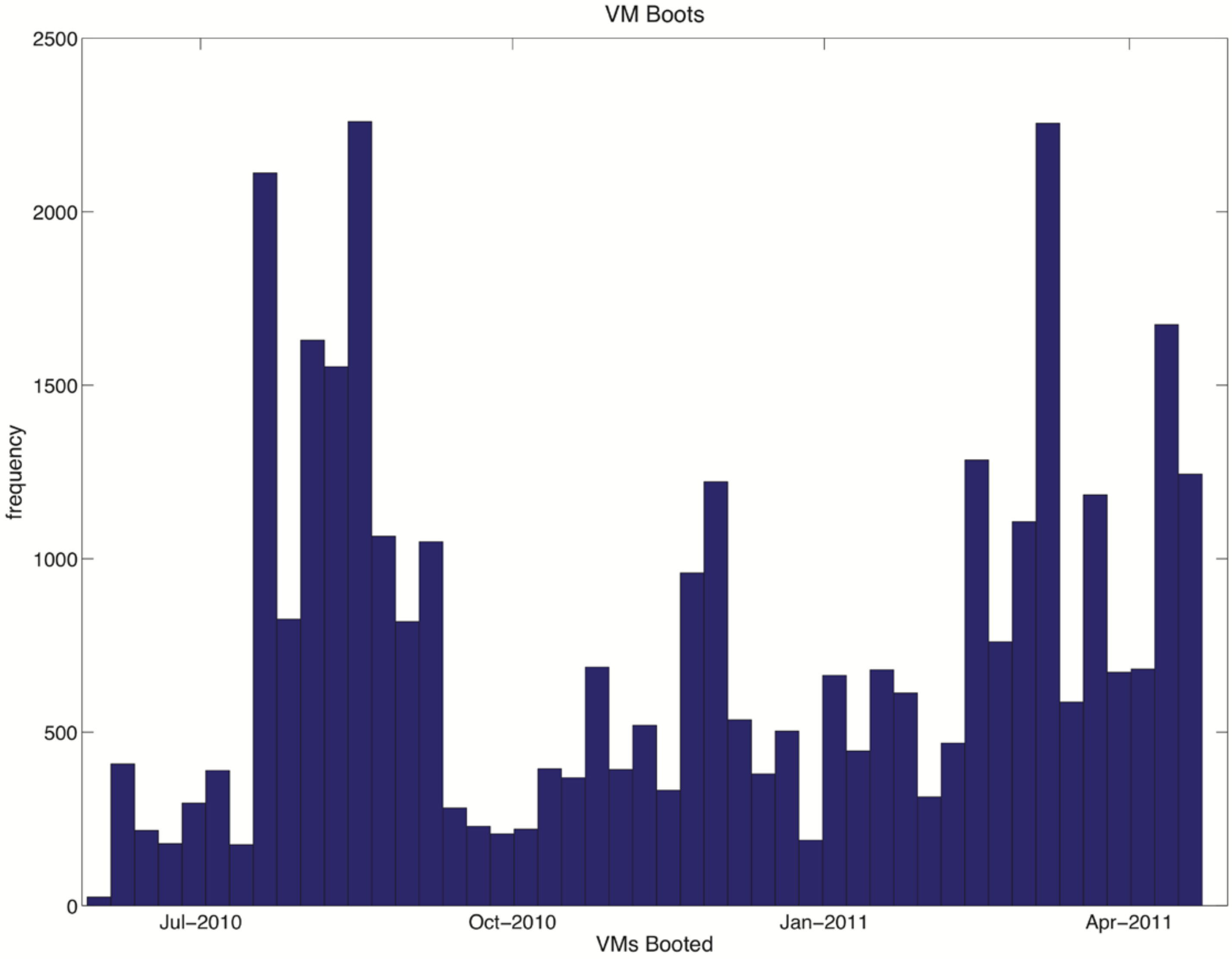
VM Run Times (CANFAR)



n = 32572

Max allowed VM Run time (1 week)

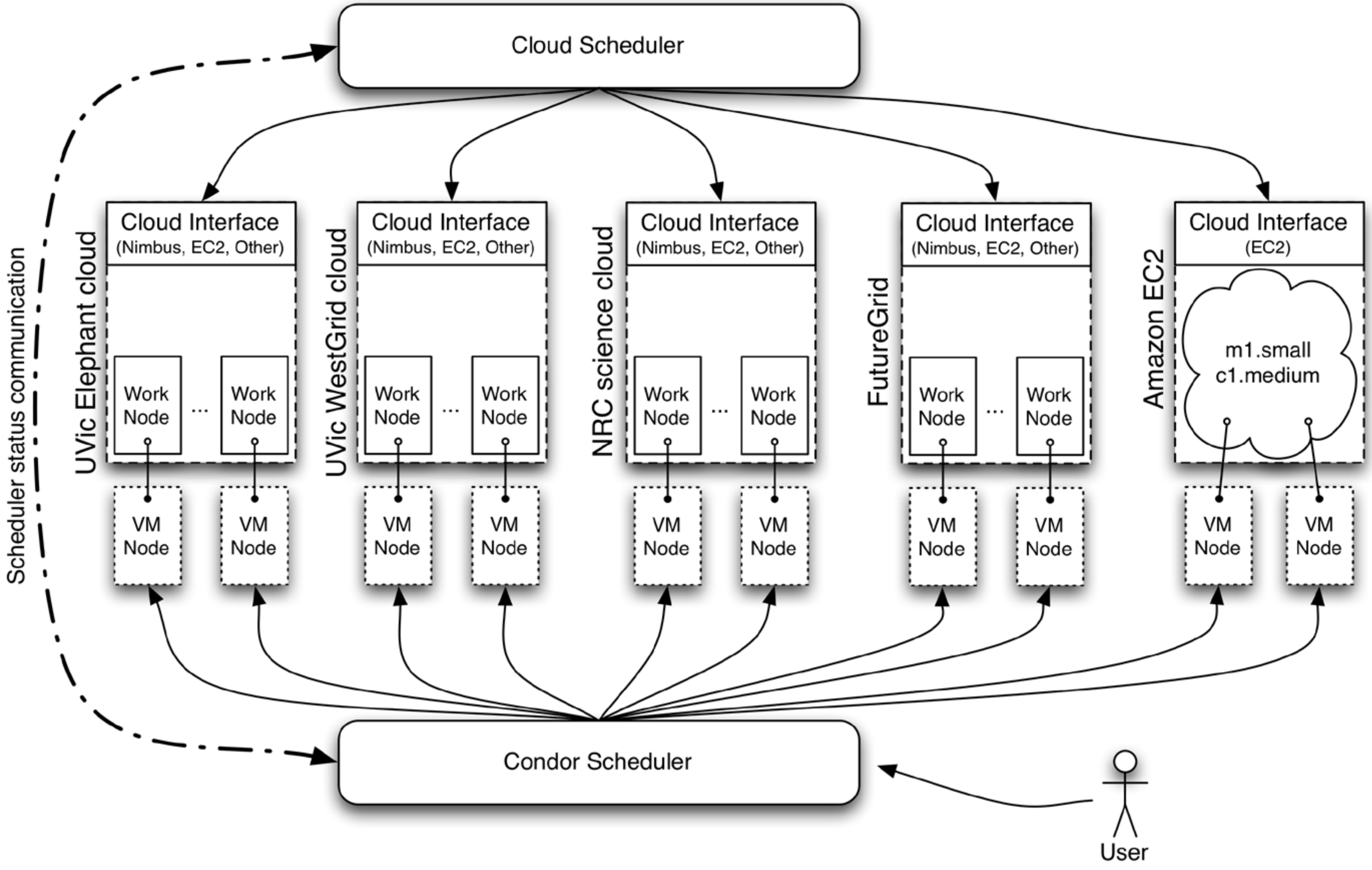
VM Boots (CANFAR)



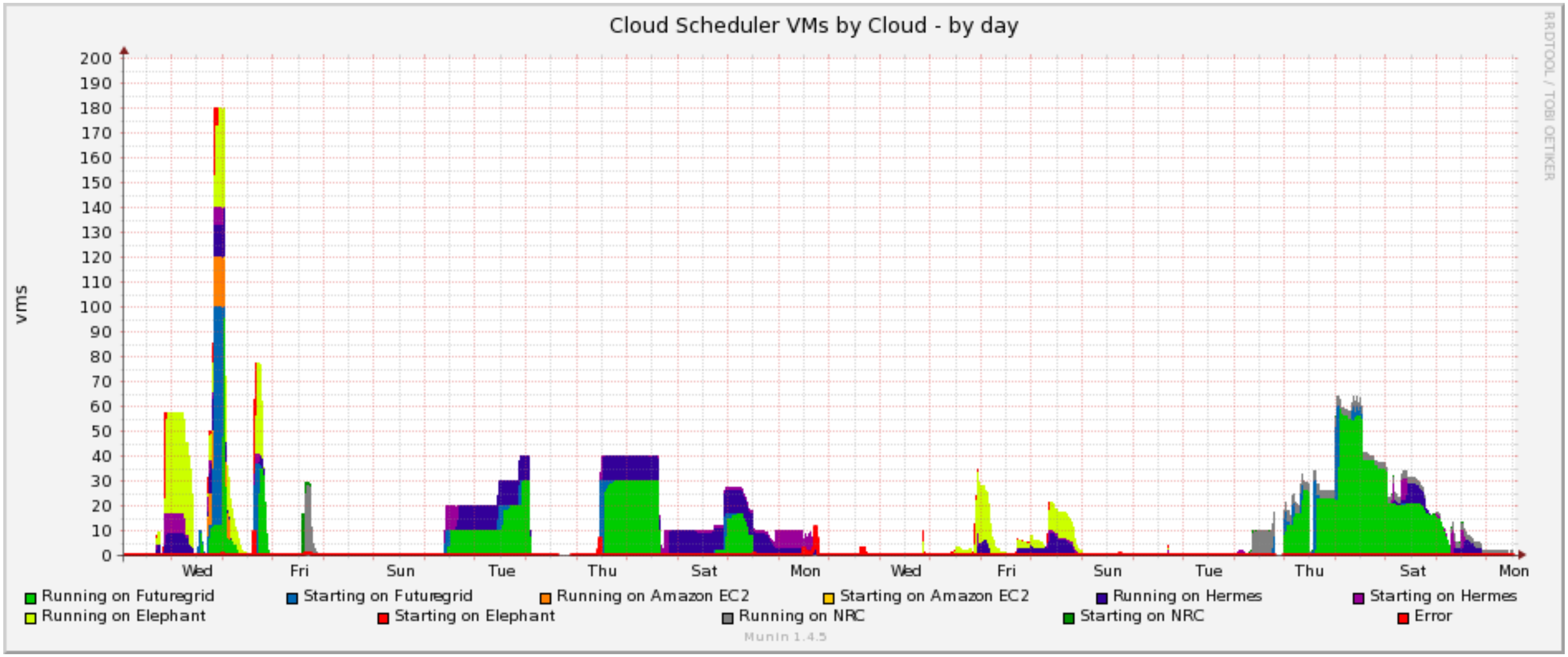
Experimental BaBar Cloud Resources

Resource	Cores	Notes
FutureGrid @Argonne Lab	100 Cores Allocated	Resources allocation to support BaBar
Elephant Cluster @UVic	88 Cores	Experimental cloud cluster hosts (xrootd for cloud)
NRC Cloud in Ottawa	68 Cores	Hosts VM image repository (repoman)
Amazon EC2	Proportional to $	Grant funding from Amazon
Hermes Cluster @Uvic	Variable (280 max)	Occasional Backfill access

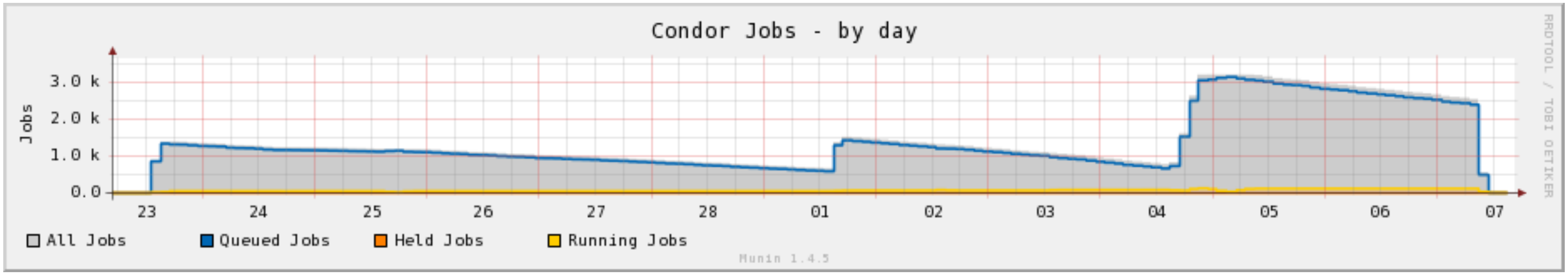
BaBar Cloud Configuration

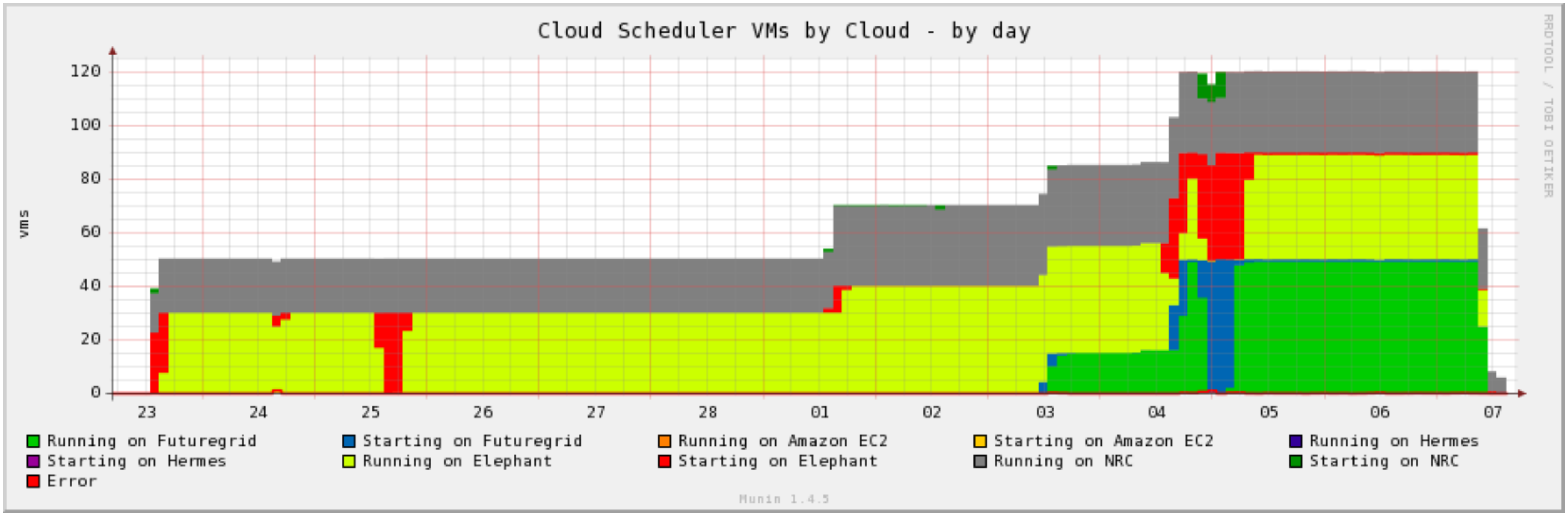


A Typical Week (Babar)

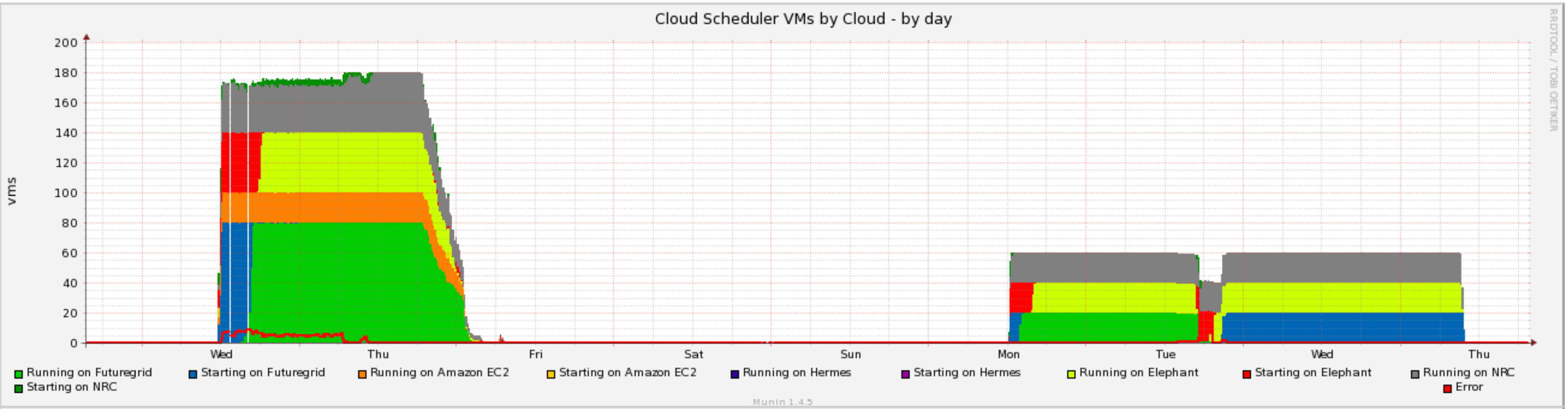


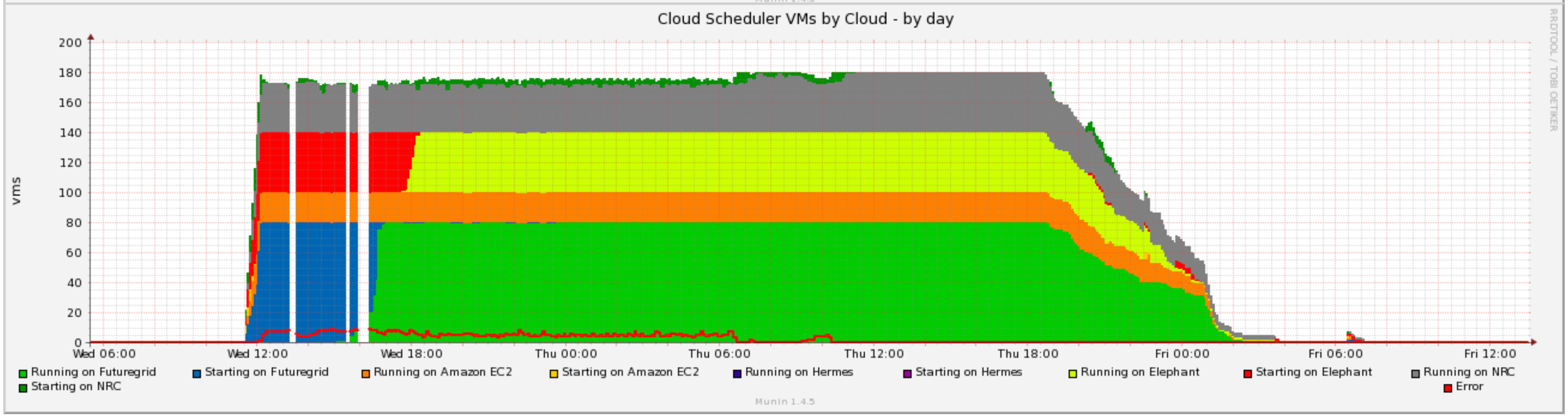
BaBar MC production



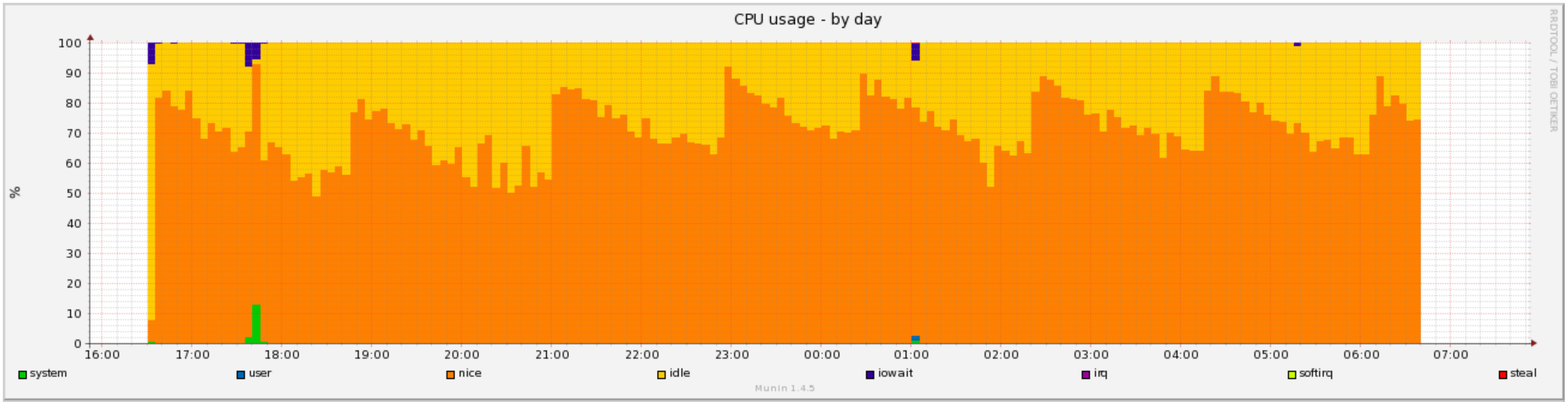


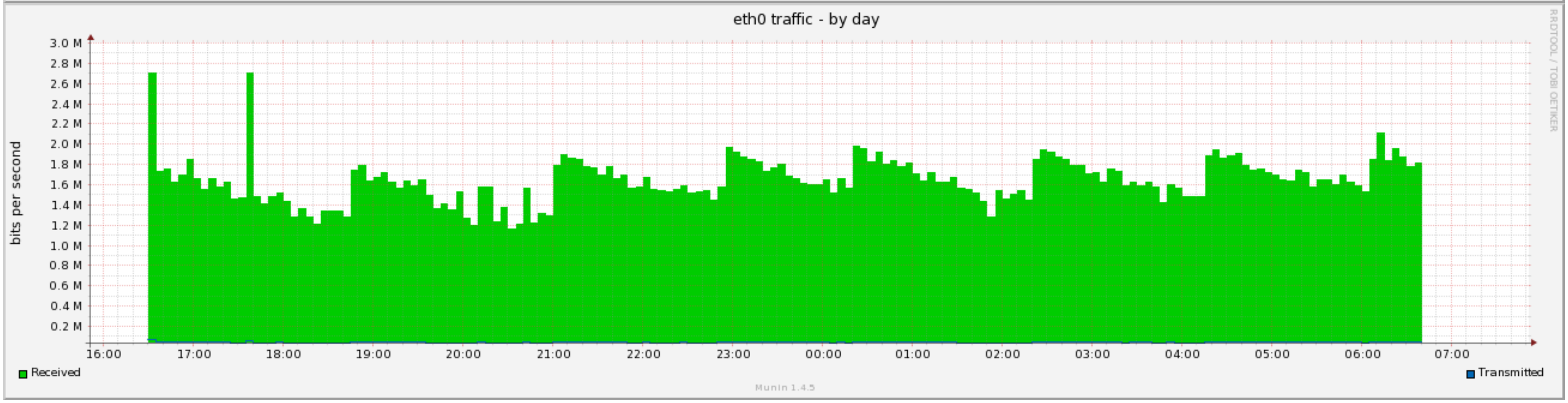
Other Examples



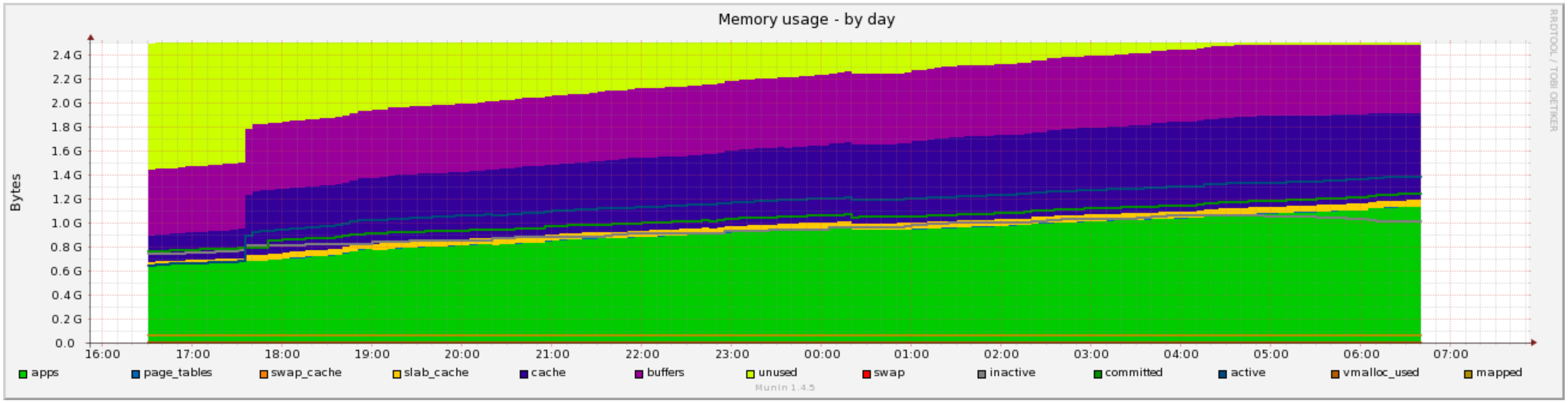


Inside a Cloud VM

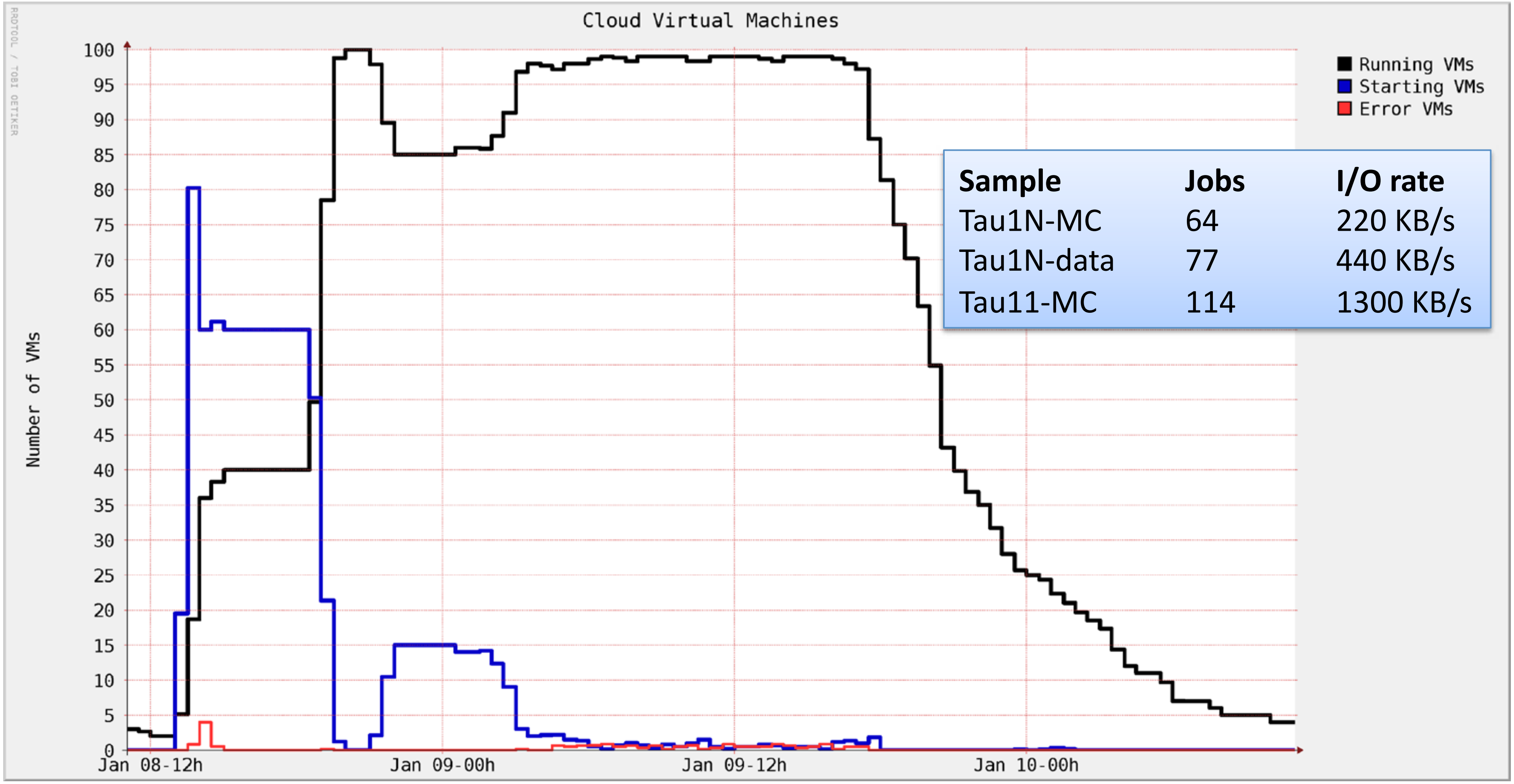




Inside a Cloud VM Cont.

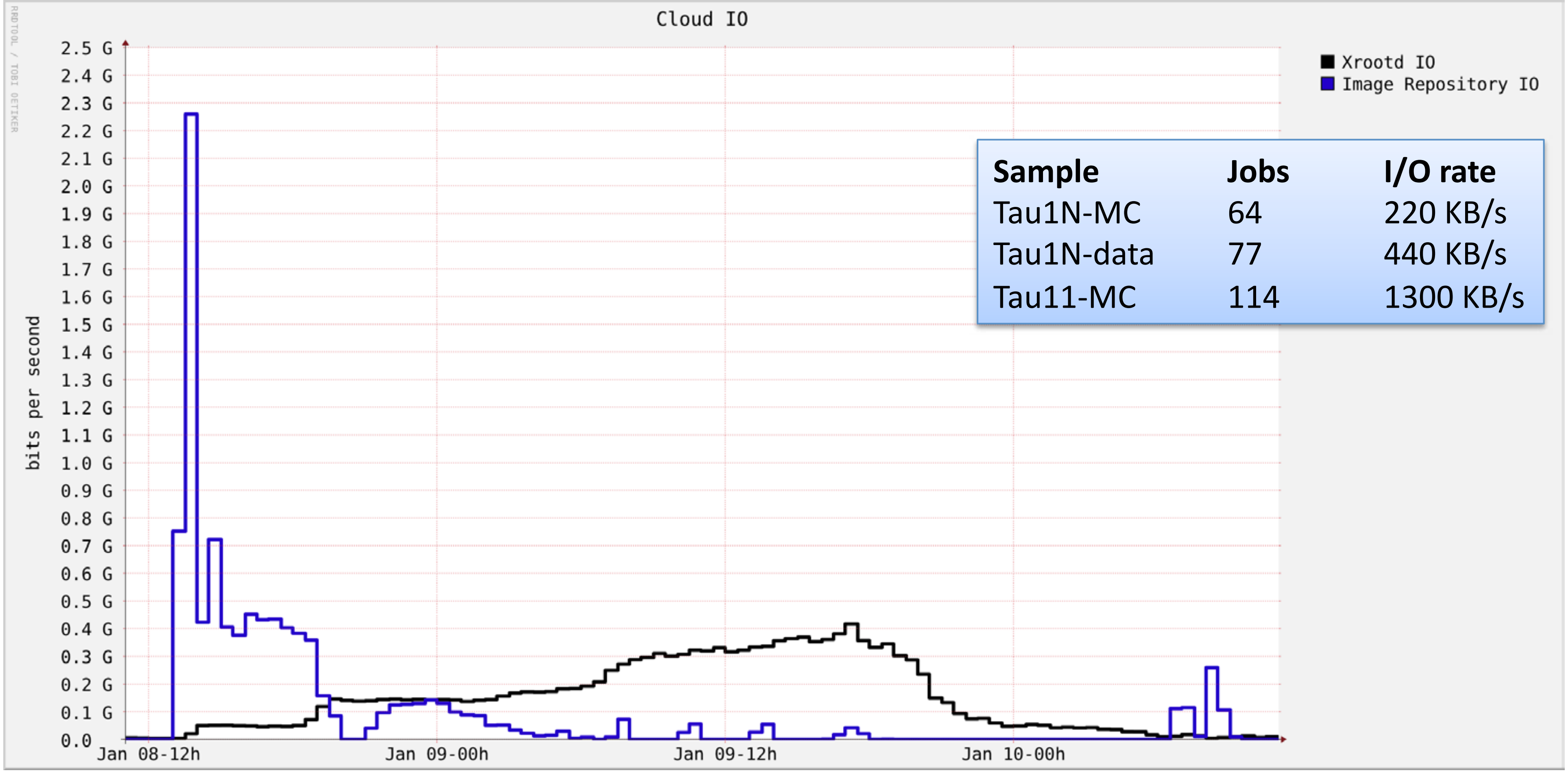


A batch of User Analysis Jobs



Sample	Jobs	I/O rate
Tau1N-MC	64	220 KB/s
Tau1N-data	77	440 KB/s
Tau11-MC	114	1300 KB/s

Cloud I/O for BaBar User Anaysis



Sample	Jobs	I/O rate
Tau1N-MC	64	220 KB/s
Tau1N-data	77	440 KB/s
Tau11-MC	114	1300 KB/s

Some Lessons Learned

- Monitoring cloud resources is difficult
  - Can you even expect the same kind of knowledge?
- Debugging user VM problems is hard for users, and hard for support
  - What do you do when the VM network doesn't come up.
- No two EC2 API implementations are the same
  - Nimbus, OpenNebula, Eucalyptus all different
- Users nicely insulated from cloud failures. If the VM doesn't come but the job doesn't get drained.

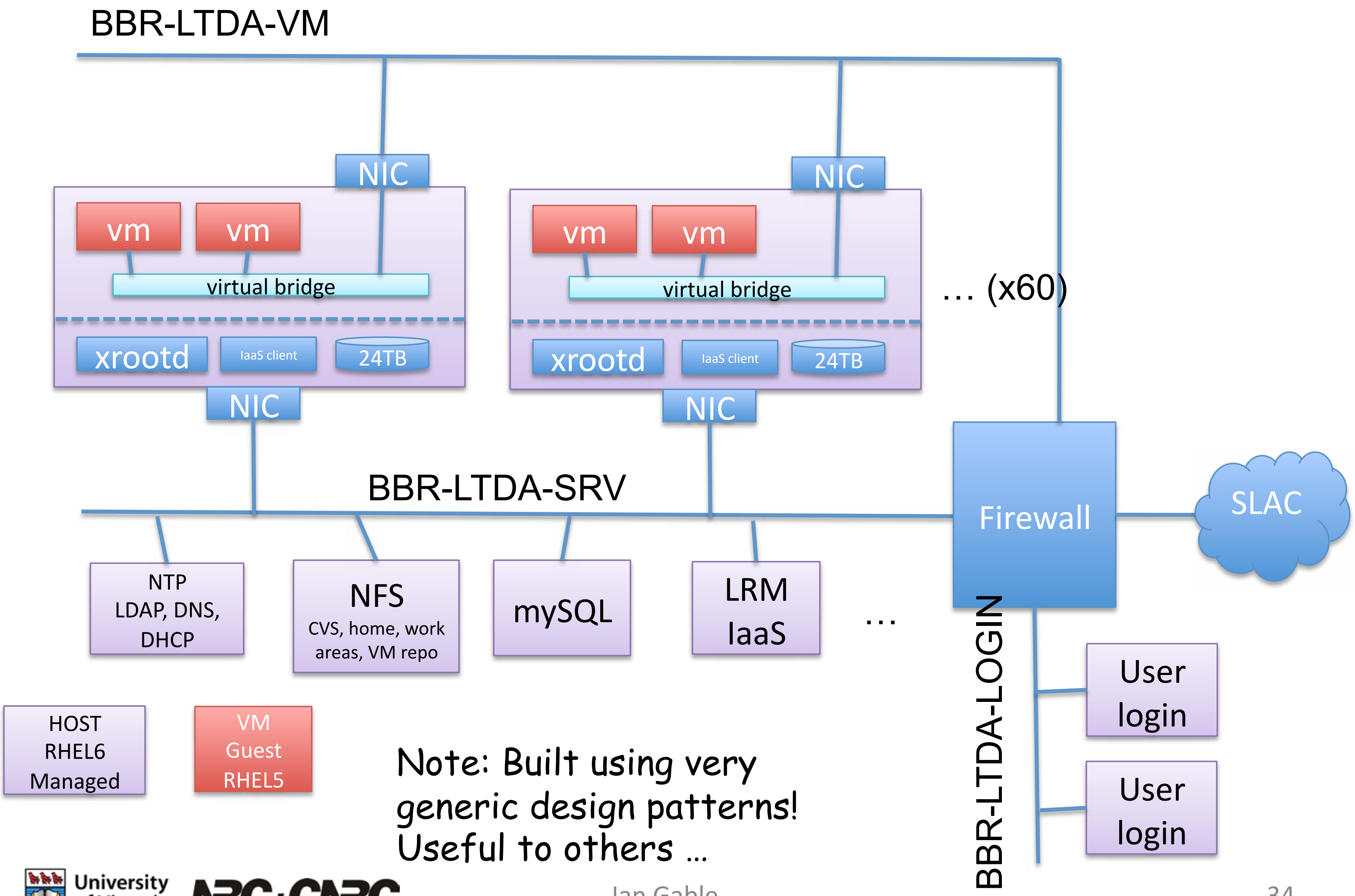
SLAC activities

Cloud in a Box:

- LTDA Analysis cloud
- The idea is to build a secure cloud to run obsolete operating systems without compromising the base OS.
- VMs are on a separate vlan, and strict firewall rules are in place.
- Users are managed through ldap on an up-to-date system.
- Uses Condor / Cloud Scheduler / Nimbus for IaaS.

SLAC Team: Homer Neal, Tina Cartaro, Steffen Luitz, Len Moss, Booker Bense, Igor Gaponenko, Wiko Kroeger, Kyle Fransham

SLAC LTDA Cluster



Note: Built using very generic design patterns! Useful to others …

Future Work/Challenges

- Increasing the the scale
  - I/O scalability needs to be proven.
  - Total number of VMs.
- Security? Leverage work of HEPiX virtualization working group.
- Booting large numbers of VM quickly on research clouds.
  - copy on write images (qcow, zfs backed storage)?
  - BitTorrent Distribution?
  - Amazon does it so we can too.

About the code

```
Ian-Gables-MacBook-Pro:cloud-scheduler igable$ cat source_files |
xargs wc -l
       0 ./cloudscheduler/__init__.py
       1 ./cloudscheduler/__version__.py
     998 ./cloudscheduler/cloud_management.py
    1169 ./cloudscheduler/cluster_tools.py
     362 ./cloudscheduler/config.py
     277 ./cloudscheduler/info_server.py
    1086 ./cloudscheduler/job_management.py
       0 ./cloudscheduler/monitoring/__init__.py
      63 ./cloudscheduler/monitoring/cloud_logger.py
     208 ./cloudscheduler/monitoring/get_clouds.py
     176 ./cloudscheduler/utilities.py
      13 ./scripts/ec2contexthelper/setup.py
      28 ./setup.py
      99 cloud_resources.conf
    1046 cloud_scheduler
     324 cloud_scheduler.conf
     130 cloud_status
    5980 total
```

- Relatively small python package, lots of cloud interaction examples

http://github.com/hep-gc/cloud-scheduler

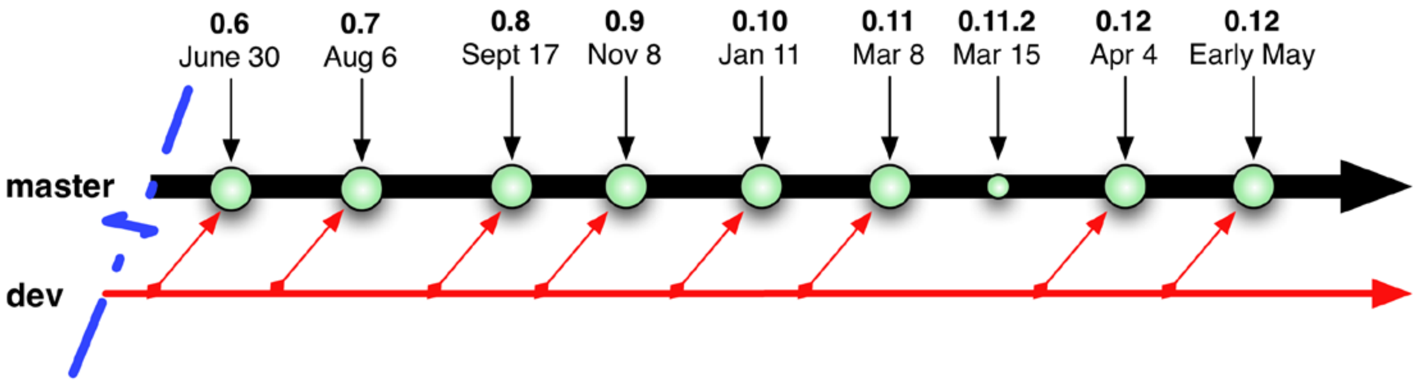
Summary

- Modest I/O jobs can be easily handled on IaaS clouds
- Early experiences are promising
- More work to show scalability
- Lots of open questions

More Information

- Ian Gable (igable@uvic.ca)
- cloudscheduler.org
- Code on GitHub:
  - http://github.com/hep-gc/cloud-scheduler
  - Run as proper open source project

Roadmap



Acknowledgements

canarie

Canada's Advanced Research and Innovation Network
Le réseau évolué de recherche et d'innovation du Canada

Future Grid

University of Victoria

NRC·CNRC

CANFAR

Canadian Advanced Network for Astronomical Research

amazon web services™

Start of extra slides

CANFAR

- CANFAR needs to provide computing infrastructure for 6 astronomy survey projects:

Survey		Lead	Telescope
Next Generation Virgo Cluster Survey	NGVS	UVic	CFHT
Pan-Andromeda Archaeological Survey	PAndAS	UBC	CFHT
SCUBA-2 All Sky Survey	SASSy	UBC	JCMT
SCUBA-2 Cosmology Legacy Survey	CLS	UBC	JCMT
Shapes and Photometric Redshifts for Large Surveys	SPzLS	UBC	CFHT
Time Variable Sky	TVS	UVic	CFHT

CFHT: Canada France Hawaii Telescope

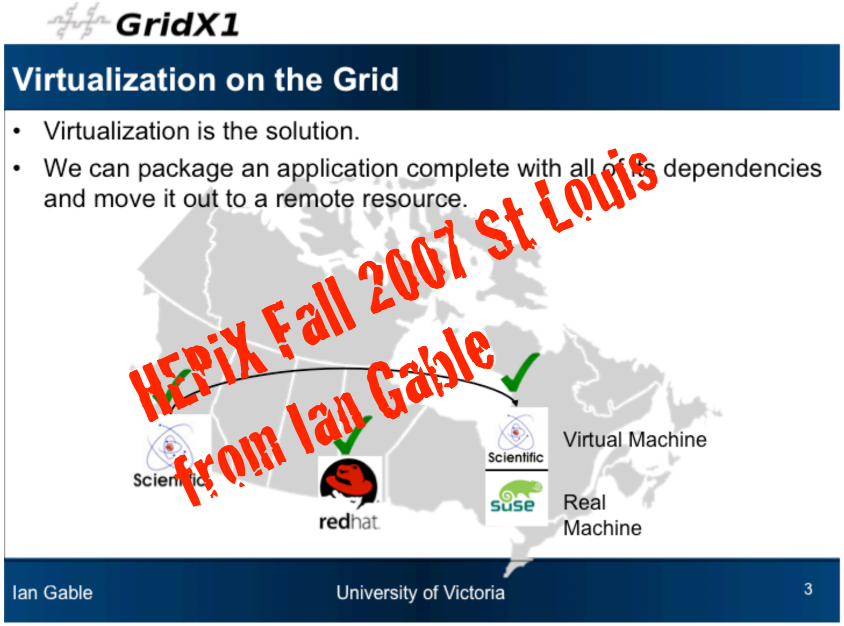
JCMT: James Clerk Maxwell Telescope

Cloud Scheduler Goals

- Don’t replicate existing functionality.
- To be able to use existing IaaS and job scheduler software together, **today.**
- Users should be able to use the familiar HTC tools.
- Support VM creation on Nimbus, OpenNebula, Eucalyptus, and EC2, i.e. all likely IaaS resources types people are likely to encounter.
- Adequate scheduling to be useful to our users
- Simple architecture

We have been interested in virtualization for some time.

- Encapsulation of Applications
- Good for shared resources
- Performs well as shown at HEPiX



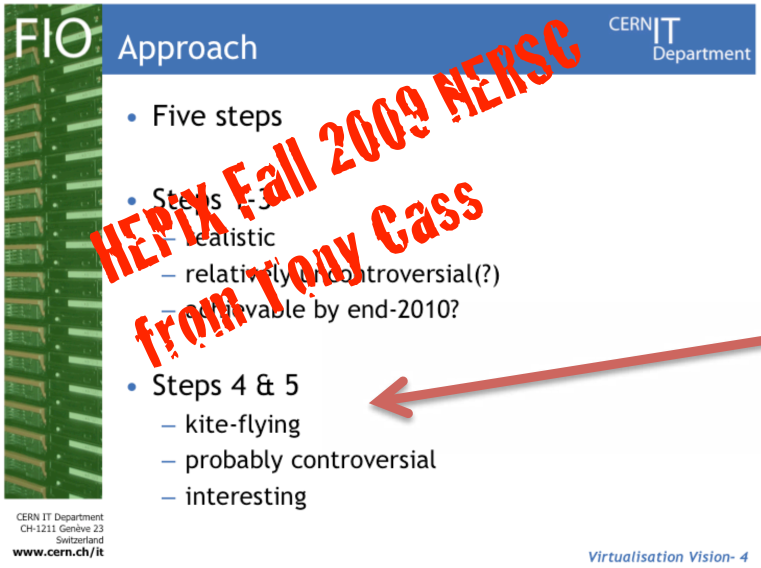
GridX1

Virtualization on the Grid

- Virtualization is the solution.
- We can package an application complete with all of its dependencies and move it out to a remote resource.

Ian Gable University of Victoria 3

HEPiX Fall 2007 St Louis from Ian Gable



FIO

Approach

CERN IT Department

- Five steps
- Steps 1-3
  - realistic
  - relatively uncontroversial(?)
  - achievable by end-2010?
- Steps 4 & 5
  - kite-flying
  - probably controversial
  - interesting

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Virtualisation Vision- 4

HEPiX Fall 2009 NERSC from Tony Cass

We are interested in pursuing user provided VMs on Clouds. These are steps 4 and 5 as outlined it Tony Cass' "Vision for Virtualization" talk at HEPiX NERSC.