An Adaptive Batch Environment for Clouds

Ian Gable

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

University of Victoria, Victoria, Canada

National Research Council of Canada, Ottawa

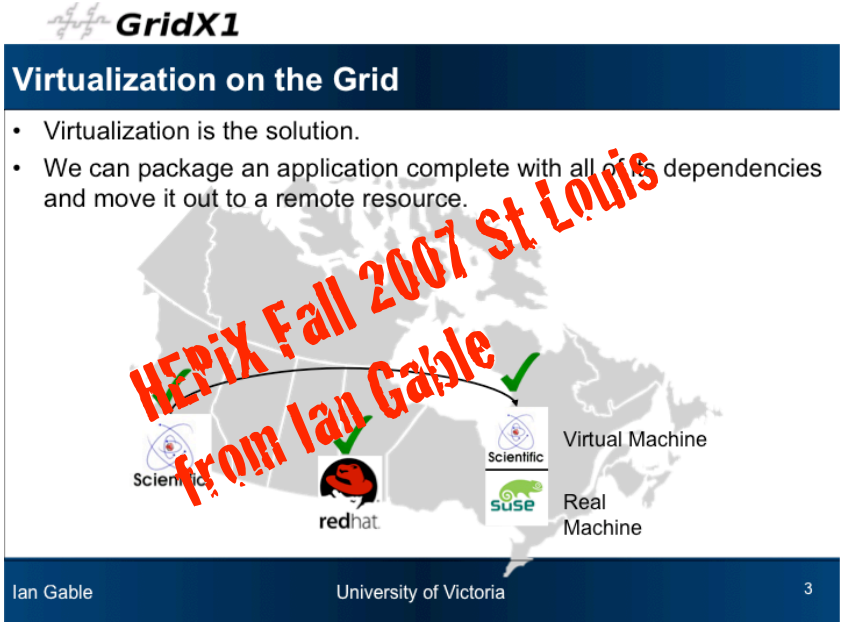
HEPiX Spring 2010, Lisbon

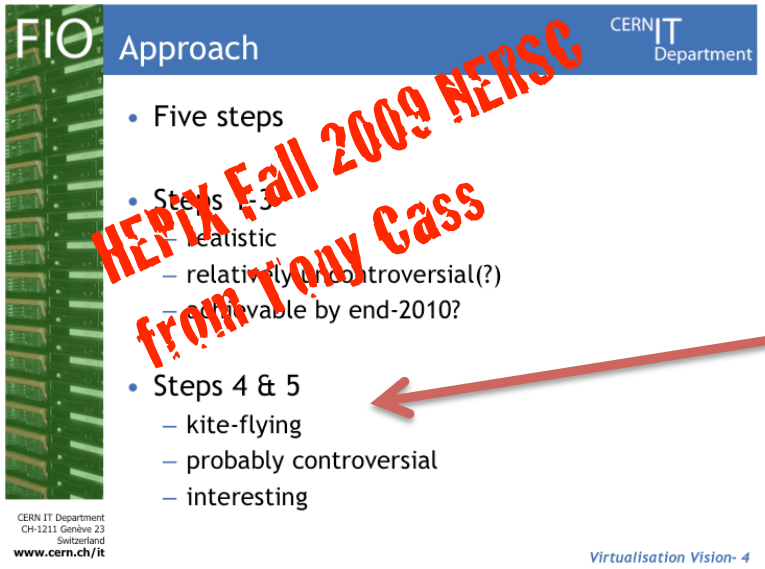
Outline

- History and HEPiX Context for this Talk
- Motivation
  - HEP Legacy Data Project
  - CANFAR: Observational Astronomy
  - SAFORAH: Forestry project (not detailed today)
- Design and Implementation
- Early experiences
- Challenges and Future Work
- Cloud Scheduler Test Drive

We have been interested in virtualization for some time.

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





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.

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)
  - Forestry Earth Observation Satellite Data Project (SAFORAH)
- We expect an increasing number of IaaS clouds to be available for research computing.

HEP Legacy Data Project

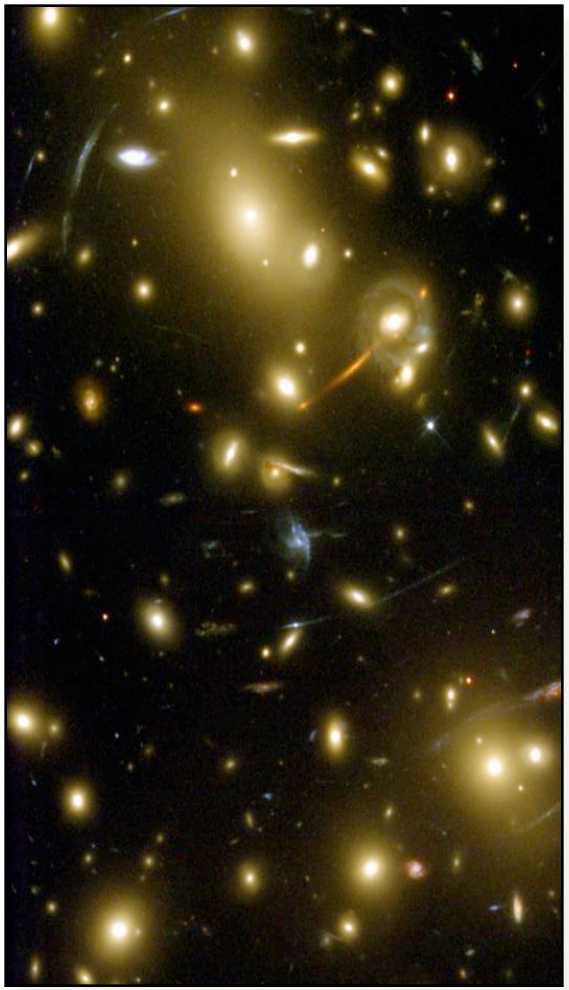
- We have been funded in Canada to investigate a possible solution for analyzing BaBar data for the next 5-10 years.
- Collaborating with SLAC who are also pursuing this goal.
- We are exploiting VMs and IaaS clouds.
- Assume we are going to be able run BaBar code in a VM for the next 5-10 years.
- We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

- 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

Virtualization:

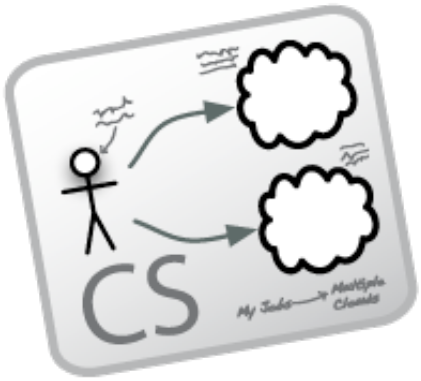
- Create Virtual Machines with these applications installed
- Run jobs for these projects on these VMs
- Users can customize the VMs to suit their specific needs

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
- Sun Grid Engine Submission to Amazon EC2
  - Release 6.2 Update 5 can work with EC2
  - Only supports Amazon
- Other solutions?



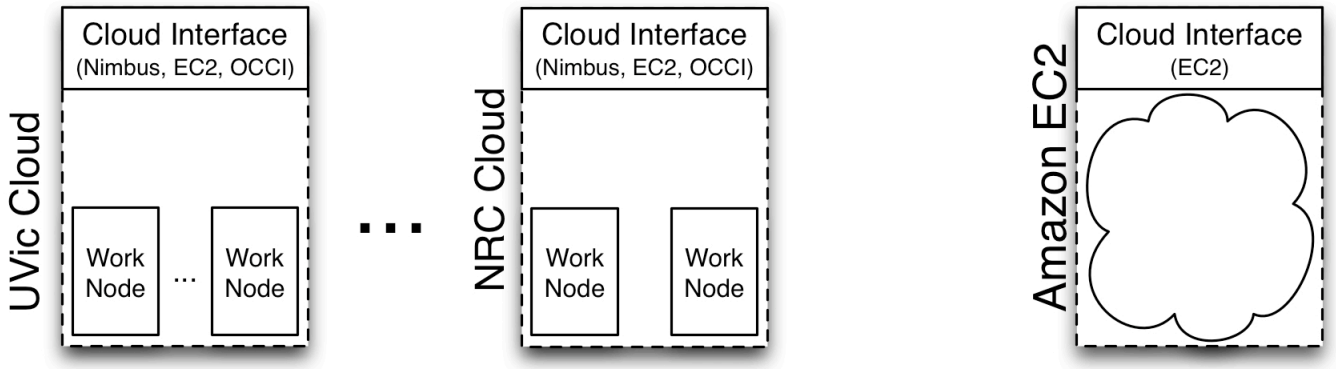
Our Solution: Cloud Scheduler

- 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

Cloud Scheduler Goals

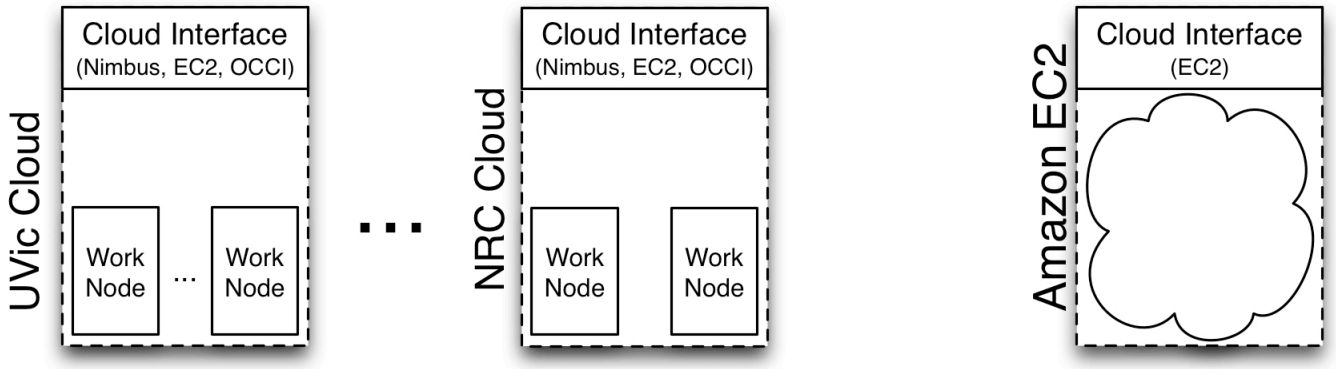
- Don't replicate existing functionality.
- To be able to use existing IaaS and job scheduler software together, **today.**
- Users should be able to use the familiar HTC tools.
- Support VM creation on Nimbus, OpenNebula, Eucalyptus, and EC2, i.e. all likely IaaS resources types people are likely to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

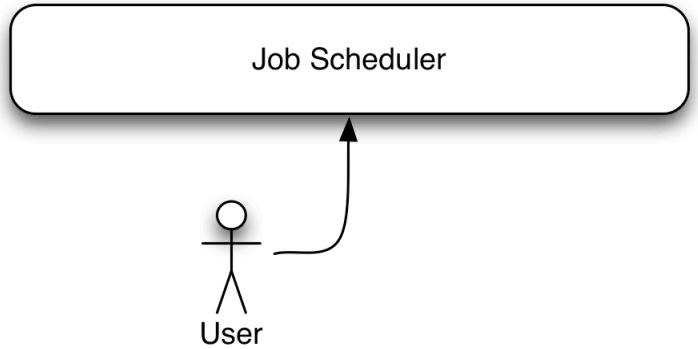
Step 1



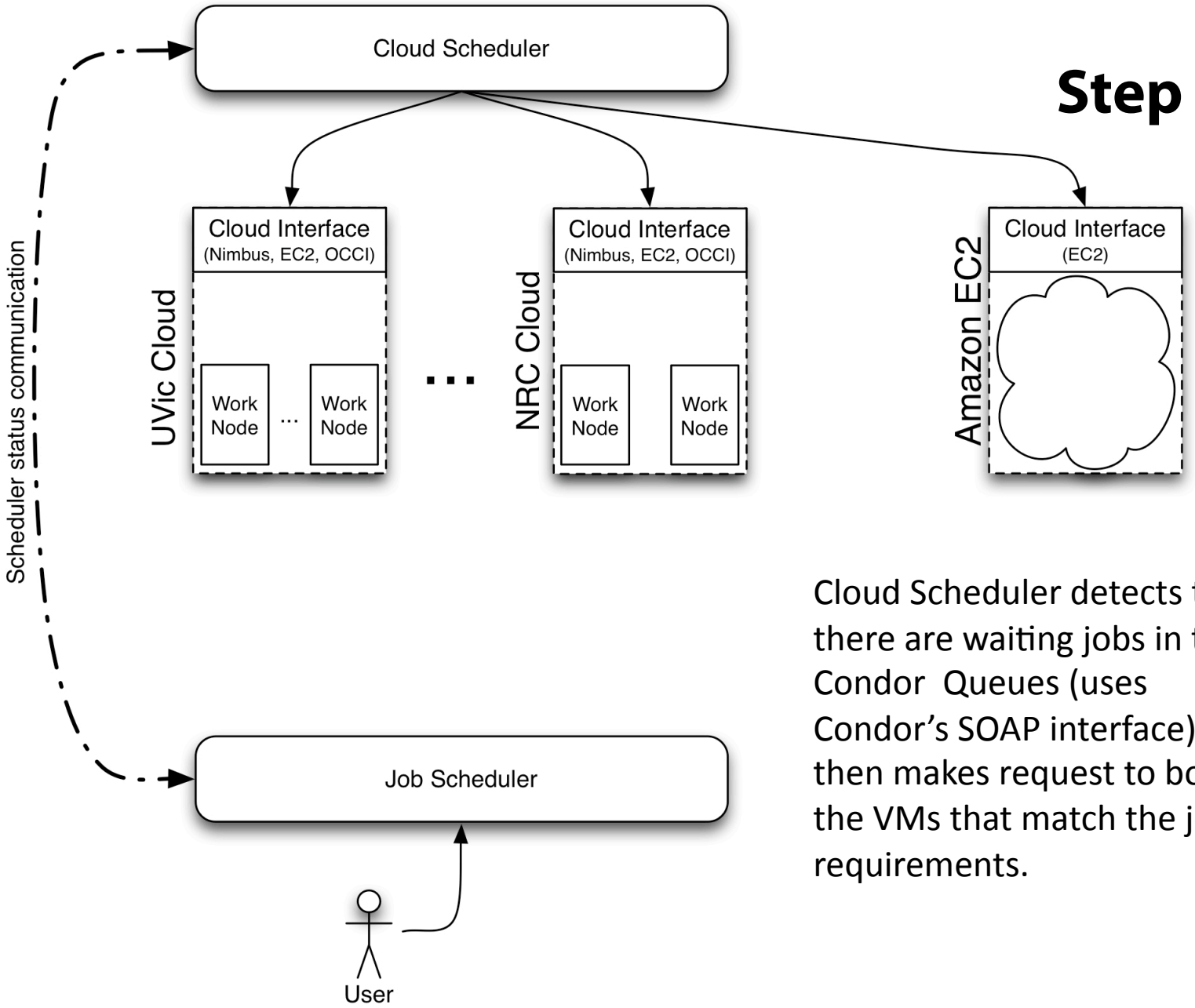
Research and Commercial clouds made available with some cloud-like interface.

Step 2

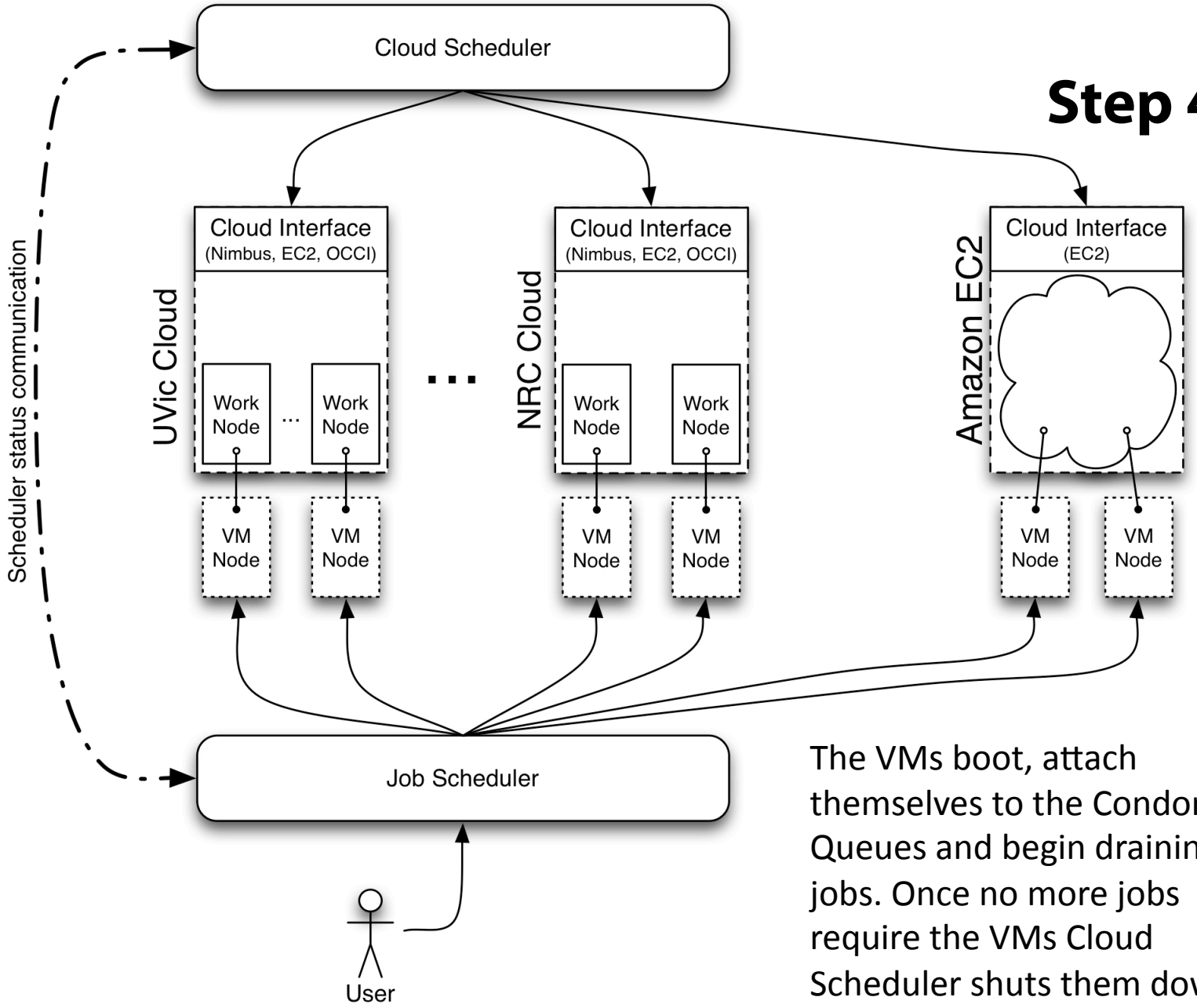




User submits to Condor Job scheduler that has no resources attached to it.



Cloud Scheduler detects that there are waiting jobs in the Condor Queues (uses Condor's SOAP interface) and then makes request to boot the VMs that match the job requirements.



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
  - Has a good SOAP API for communication
- Use OpenVPN to use clouds which only have private networking available
- 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

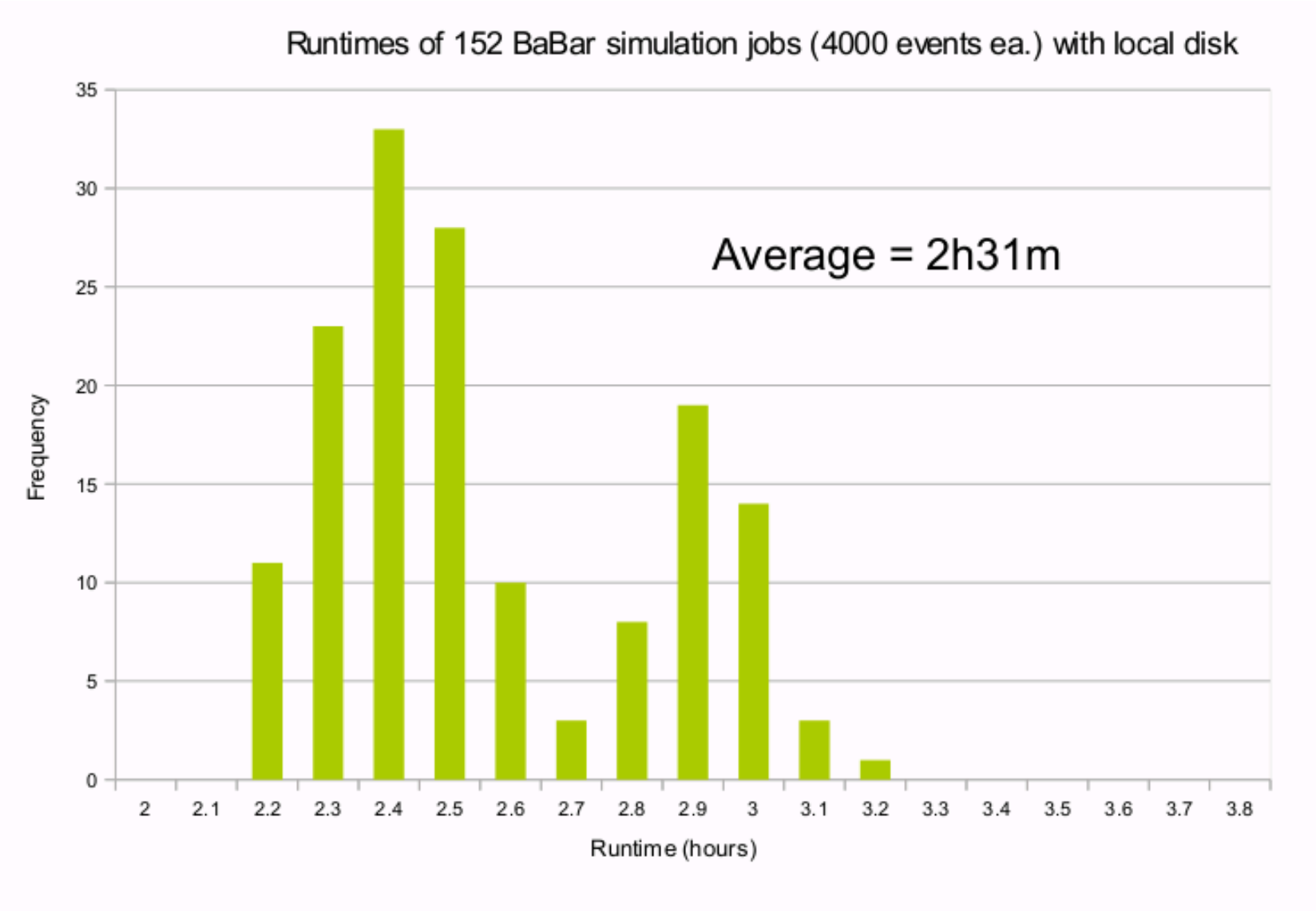
Condor Job Description File

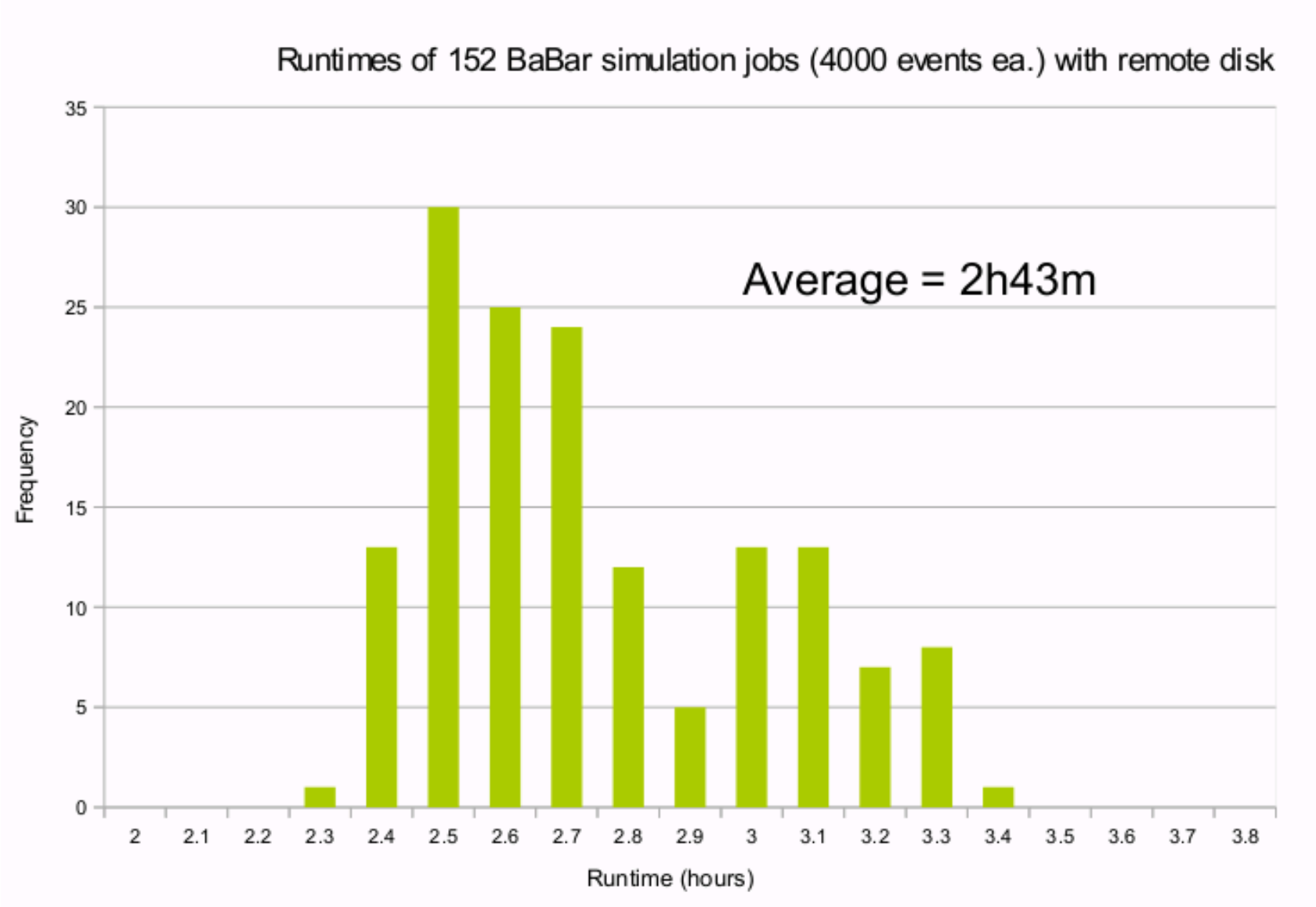
```
Universe = vanilla
Executable = red.sh
Arguments = W3-3+3 W3%2D3%2B3
Log = red10.log
Output = red10.out
Error = red10.error
should_transfer_files = YES
when_to_transfer_output = ON_EXIT

# 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
```

Early Experiences

- Nimbus deployed at 3 sites in Canada
  - One purpose built cloud development cluster; 11 Nodes (UVic):
    - VM hosting/ Cloud Storage machines, using Xen+ Lustre Kernel.
  - NRC Herzberg Institute (Victoria), 10 nodes
  - NRC Sussex (Ottawa), 6 nodes
- Test deployments of OpenNebula and Eucalyptus
- Performed successful BaBar validation

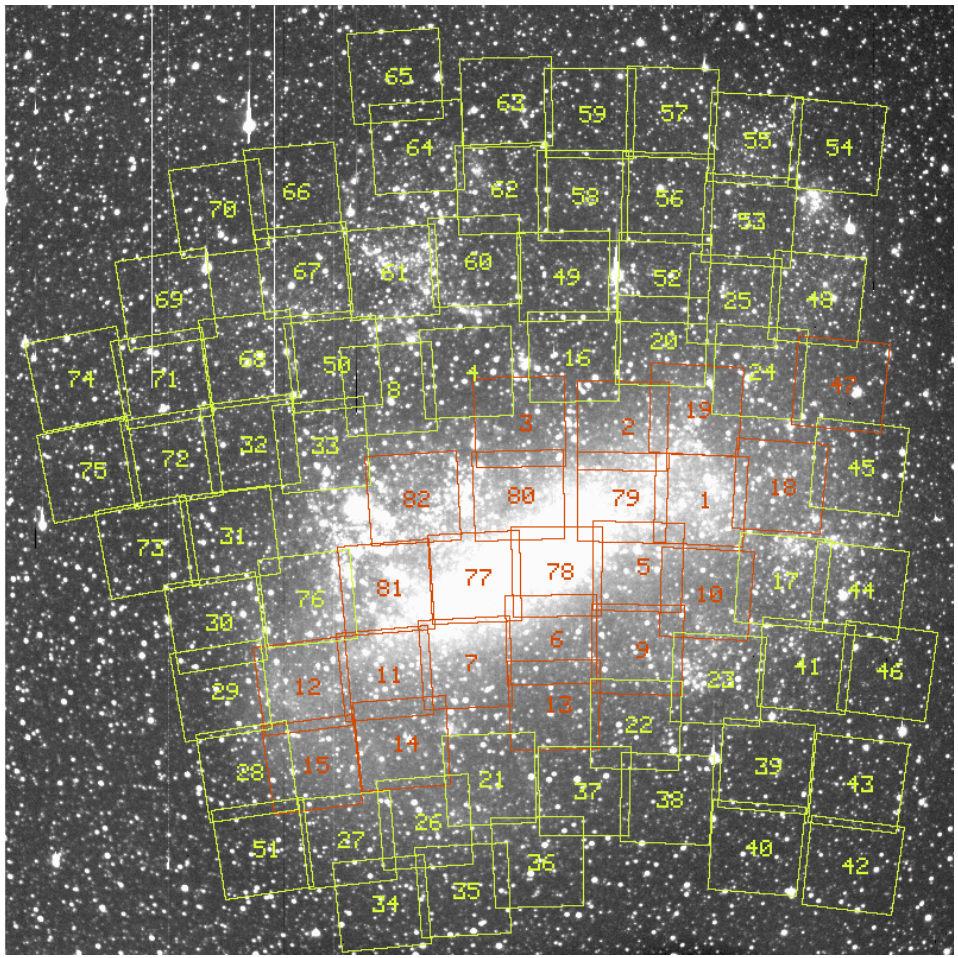
First look at cloud BaBar Simulation



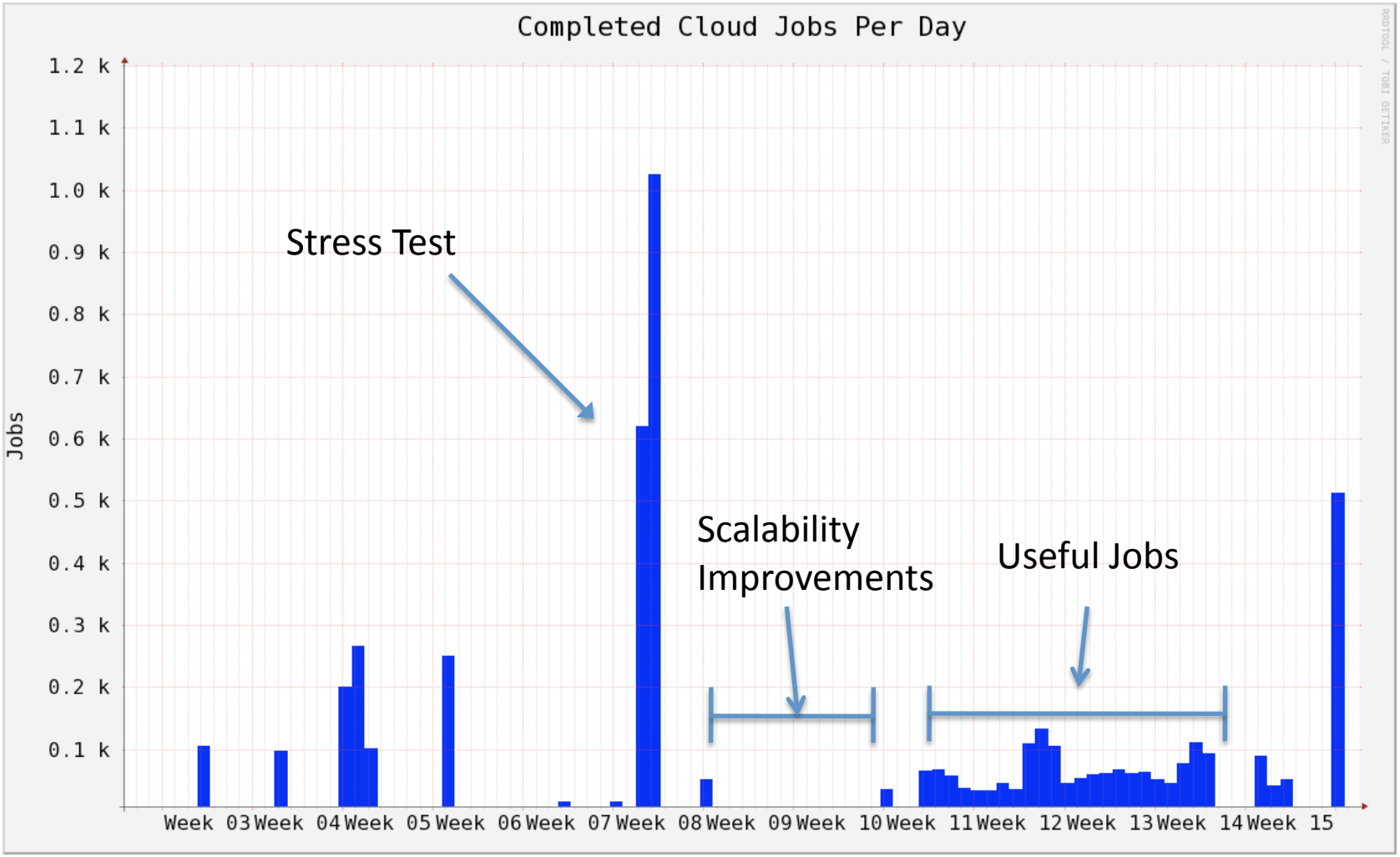

Runtimes of 152 BaBar simulation jobs (4000 events ea.) with remote disk
Average = 2h43m
Frequency
Runtime (hours)
0
5
10
15
20
25
30
35
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7
2.8
2.9
3
3.1
3.2
3.3
3.4
3.5
3.6
3.7
3.8

CANFAR: MAssive Compact Halo Objects

- 2200 useful jobs run on detailed re-analysis of data from the MACHO experiment Dark Matter search. 1% of total data set.
- 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.
- Astronomer optimistic/ happy with the environment.



Early 2010 quasi-production



Future Work/Challenges

- We are still in the (alpha) stage, so work needs to be done to ensure scalability for the workloads we expect. We haven't show the scale we need yet.
- Data Access from Cloud VMs; lots of work to be done here.
- Security assessment.
- Booting large numbers of VM quickly on research clouds.
  - copy on write images (zfs backed storage)?
  - HDFS Image Repository for Distribution?
  - BitTorrent Distribution?
  - Amazon does it so we can too.

Test Drive Cloud Scheduler

Publicly available pre-configured EC2 AMI ready to go:

```
#create a security group
$ ec2addgrp cloudscheduler -d "Used for Cloud Scheduler"
$ ec2auth cloudscheduler -P icmp -t "-1:-1"
$ ec2auth cloudscheduler -P tcp -p 22
$ ec2auth cloudscheduler -P tcp -p 40000-40050
$ ec2auth cloudscheduler -P udp -p 40000-40050
$ ec2auth cloudscheduler -P tcp -p 9618
$ ec2auth cloudscheduler -P udp -p 9618

#boot the cloud scheduler/condor VM
$ ec2run ami-f9ff1190 -k ec2-keypair -g cloudscheduler

$ ssh -i ~/.ec2/id_rsa-ec2-keypair \
  root@ec2-75-101-197-134.compute-1.amazonaws.com
[root@ec2-75-101-197-134 ~]# cat README
```

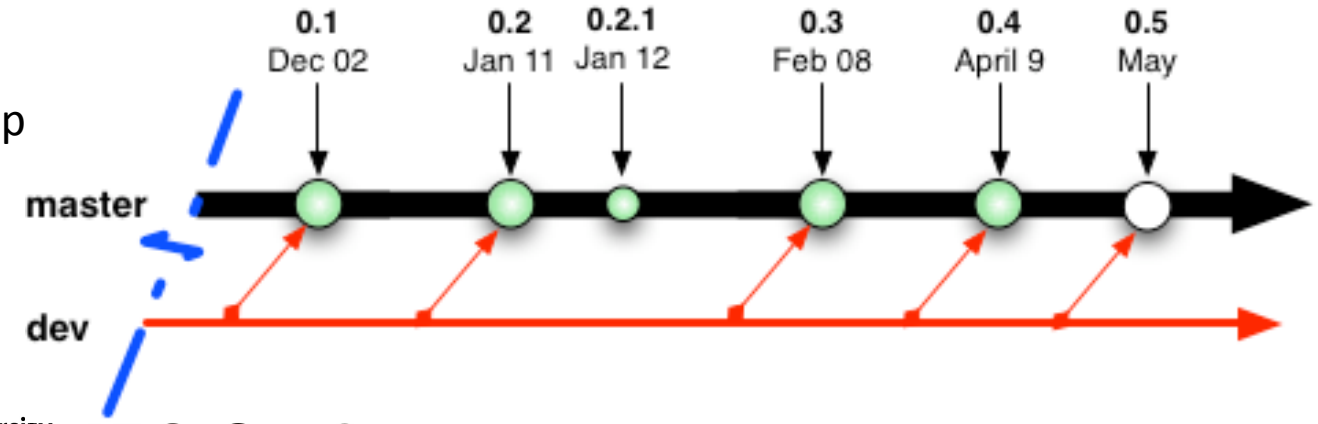
Summary

- Cloud Scheduler is a simple tool for running batch workloads on different IaaS clouds
- Early experiences are promising
- More work to show scalability
- Lots of open questions
- Try it today

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
- http://twitter.com/cloudscheduler

Roadmap



Acknowledgements

canarie

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

University of Victoria

NRC·CNRC

CANFAR

Canadian Advanced Network for Astronomical Research

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