

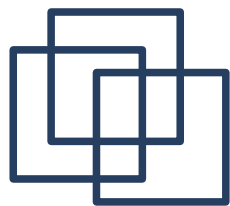


---

# Research Computing in a Distributed Cloud Environment

Kyle Fransham

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



# Outline

---

- Motivation
  - Why run science jobs on clouds?
- Implementation
  - Cloud Scheduler allows us to make use of many clouds.
- Results
  - Alpha prototype is successful.



# Why clouds?

---

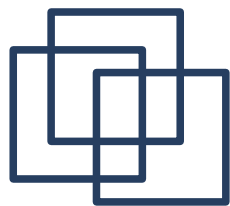
- Different scientific experiments have different requirements:

- OS, libraries, etc.



- For older experiments, older operating systems are required.

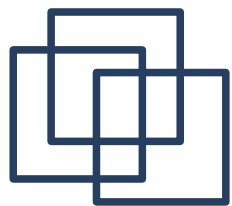




# Why Clouds?

---

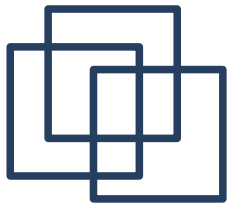
- We encapsulate the software in a Virtual Machine, and we can run many types of new and old applications on a single cluster.
- This technology (Infrastructure as a Service) already exists:
  - Amazon EC2 / Rackspace
    - Commercial providers
  - Nimbus / OpenNebula / Eucalyptus
    - Turn your cluster into a cloud.



# A Distributed Cloud

---

- Users typically want to use as many resources as are available to them, but it's a big exercise to setup their computing environment on many different clouds.
  - BUT, if the user's environment is in a VM, that VM can be booted anywhere there are free resources.
  - We're creating the piece that allows a user to submit a job as they would to a regular batch system, but have it run on many different physical sites. It's the Cloud Scheduler.



# Cloud Scheduler

- A user submits a job to a regular job scheduler using familiar batch tools.

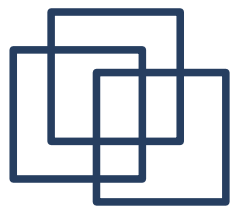
```
Executable = /home/babar/SP26.0.0/workdir/runJob.sh
Universe = vanilla
Log = condor.log
Output = moose.output
Error = moose.error
InitialDir = /home/babar/SP26.0.0/workdir
#should_transfer_files = yes
#WhenToTransferOutput = ON_EXIT

Requirements = VMType =?= "bbrSim"
+VMName = "bbrSim"
+VMLoc = "http://alto.cloud.nrc.ca/bbrSim"
+VMNetwork = "public"
+VMCPUArch = "x86"
+VMStorage = "1"
+VMCPUCores = "1"
+VMMem = "2048"
getenv = True

Queue
```

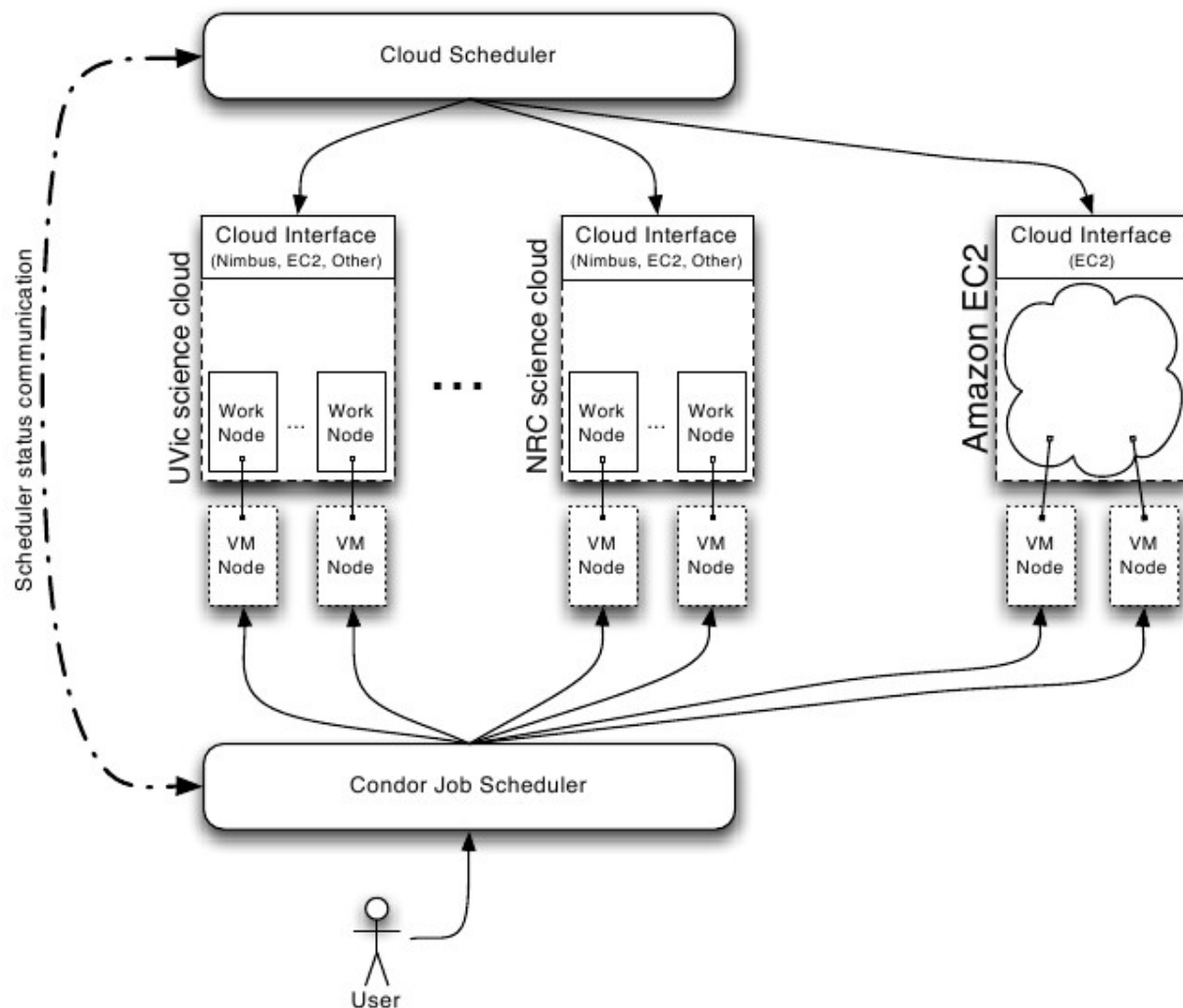
Job Scheduler

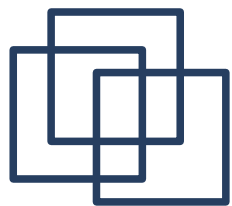
- The job scheduler has no resources to run the job yet, so the job is queued.



# Cloud Scheduler

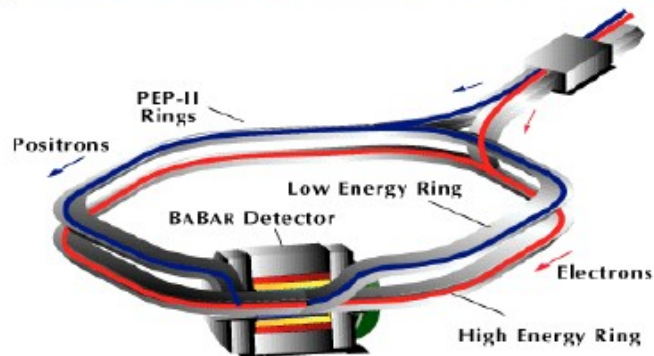
- The Cloud Scheduler knows about the IaaS resources available, and it polls the job scheduler's queue.
- When it sees a new job in the queue, the Cloud Scheduler selects a resource and sends the instruction to boot a VM.
- The VM boots, registers itself with the job scheduler, and runs the job.



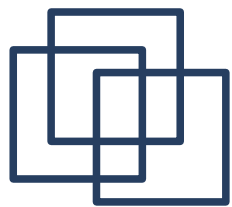


# Jobs for the Cloud

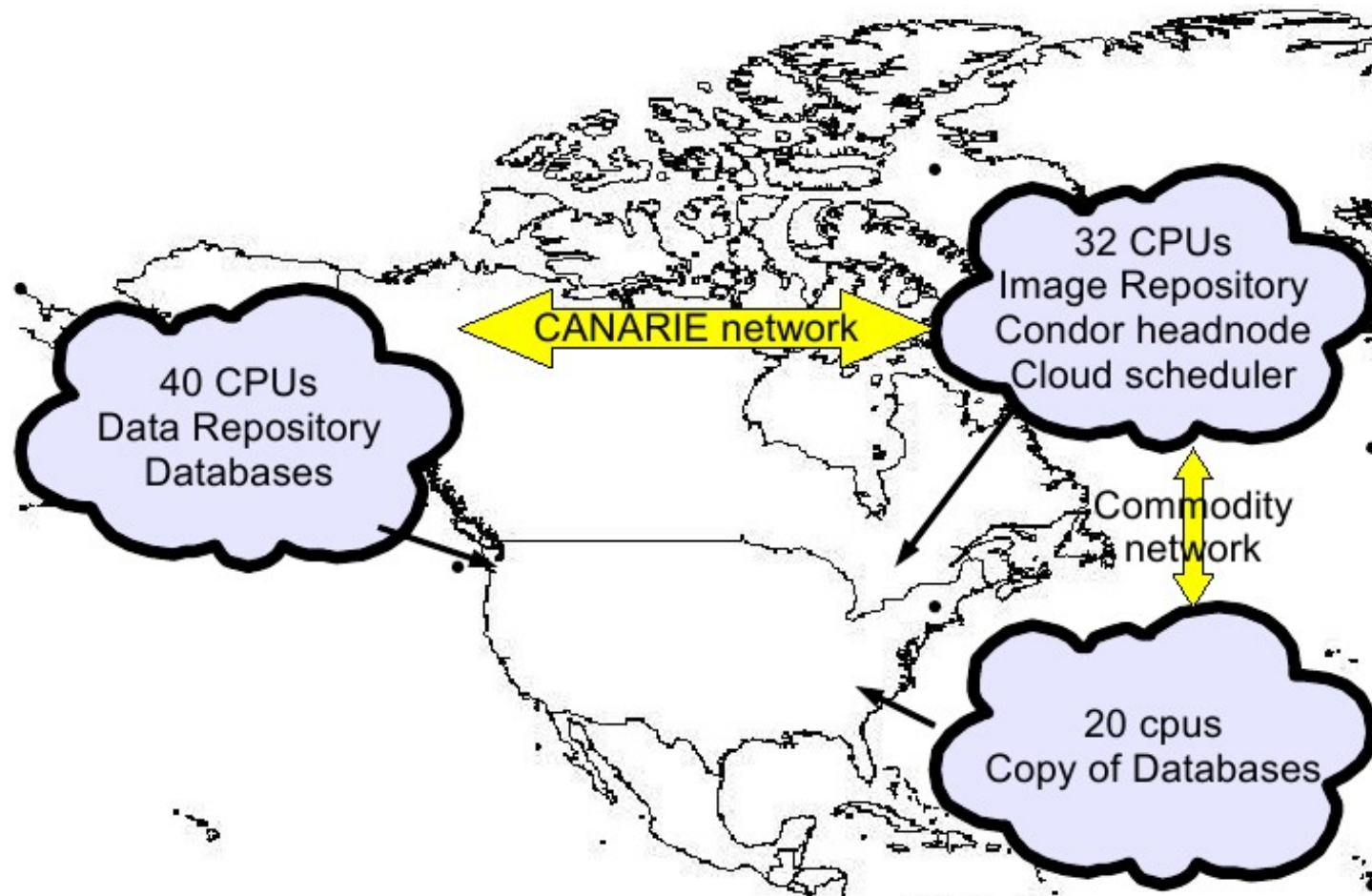
- The BaBar experiment at SLAC in Stanford, CA uses the cloud to simulate electron-positron collisions in their detector.

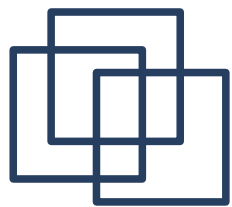






# A Distributed Cloud



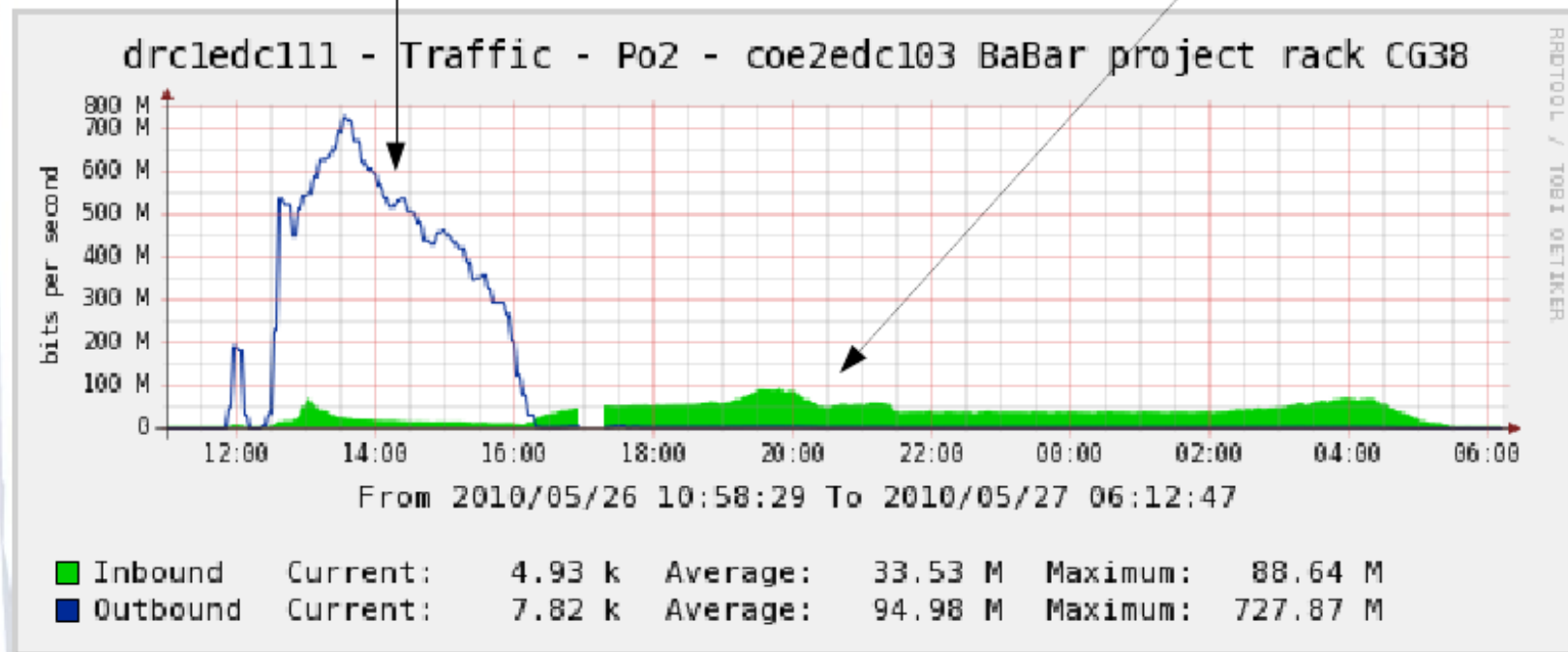


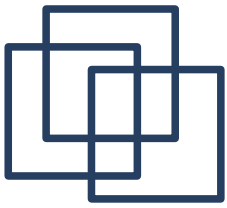
# Network Traffic (UVic - NRC)

- BaBar jobs running on the cloud.

VM images being copied  
from NRC to UVic

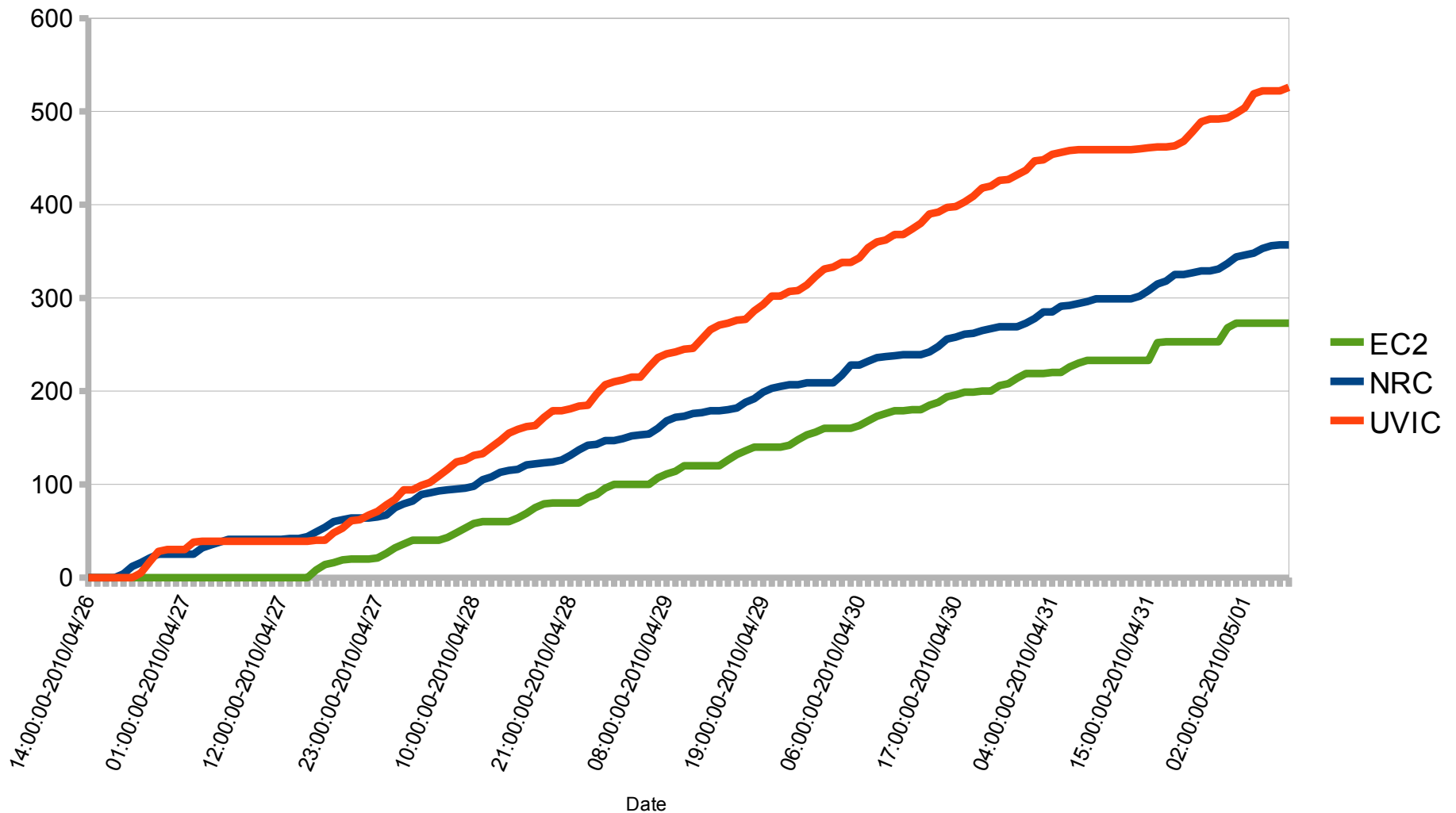
VMs running at NRC  
accessing databases and  
writing results to UVic.

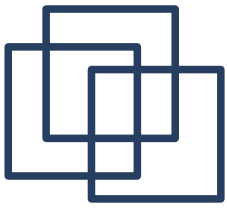




# Completed Jobs

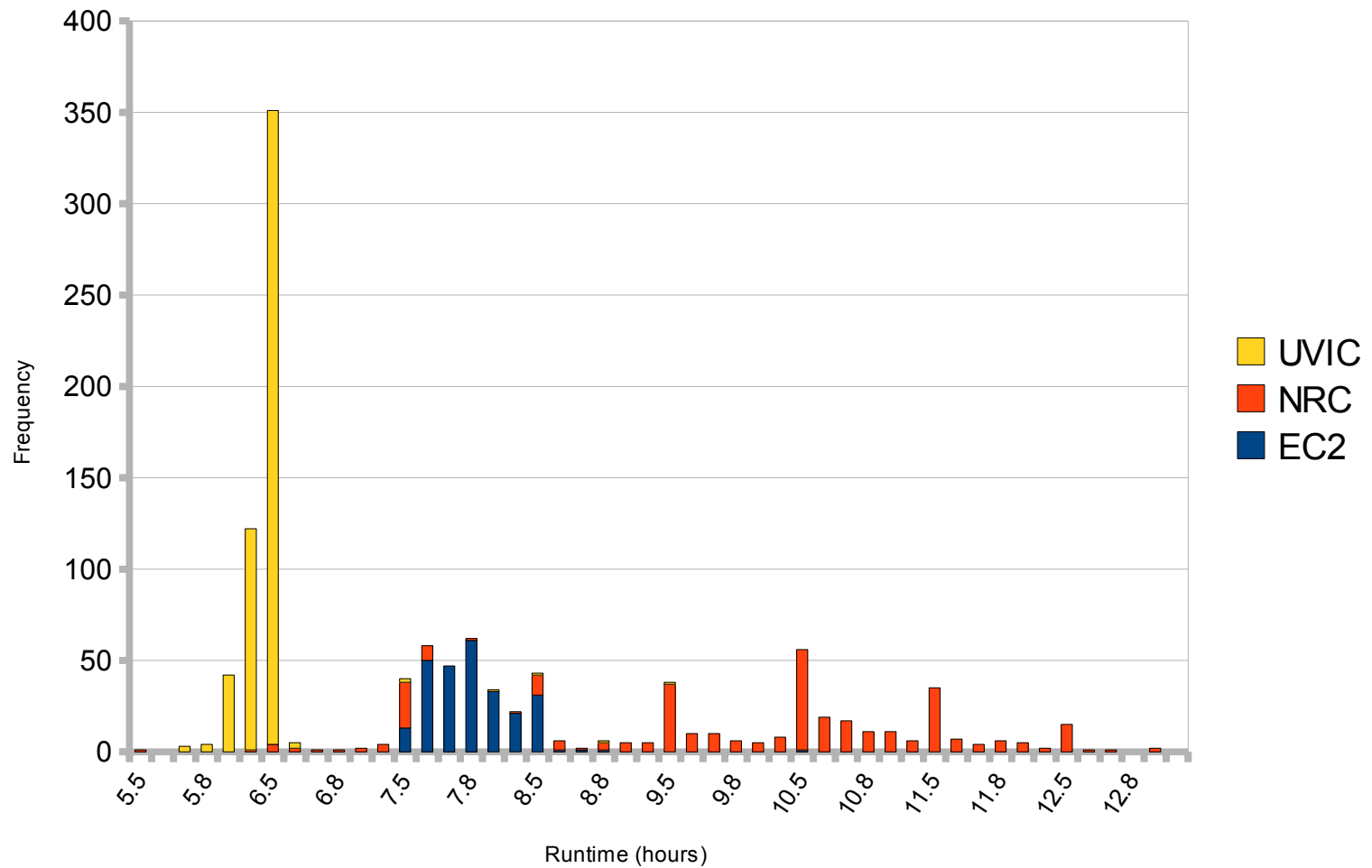
## Completed Jobs by Site

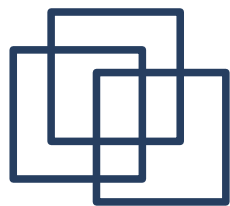




# Job Runtimes

## Job Runtimes per Site

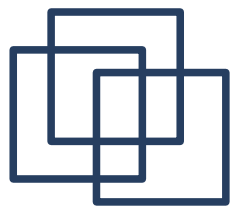




# Summary

---

- Cloud Scheduler can be used to take advantage of many clouds.
- Still in heavy development, but alpha prototype is successful.
- Currently, support is for Nimbus and EC2 clouds.
- Test it out yourself:
  - <http://wiki.github.com/hep-gc/cloud-scheduler/cloud-scheduler-test-drive>



# Acknowledgments

- Research Team:



Wayne Podaima (NRC), Roger Impey (NRC), Ashok Agarwal (UVIC),  
Duncan Penfold-Brown (UVIC), Kyle Fransham (UVIC), Ron Desmarias (UVIC), Patrick Armstrong (UVIC),  
Andre Charbonneau (NRC), Randall Sobie (UVIC), Colin Leavett-Brown(UVIC)



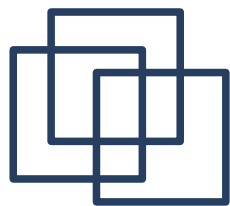
Ian Gable (UVIC)

- Funding:

canarie

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





# Links

---

- Public Webpage:
  - <http://www.cloudscheduler.org/>
- Source code / wiki / issue tracker:
  - <http://wiki.github.com/hep-gc/cloud-scheduler/>