Cloud Scheduler

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

University of Victoria NRC · CNRC

Outline

- Motivation for Cloud Scheduler
  - One high energy physics project
  - One observation astronomy project
- Design and Implementation of Cloud Scheduler
- Some preliminary results

Motivation for Cloud Scheduler

- Two projects are using Cloud Scheduler
- The Canadian Advanced Network for Astronomical Research (CANFAR)
- The High Energy Physics Legacy Data project

High Energy Physics Legacy Data Project

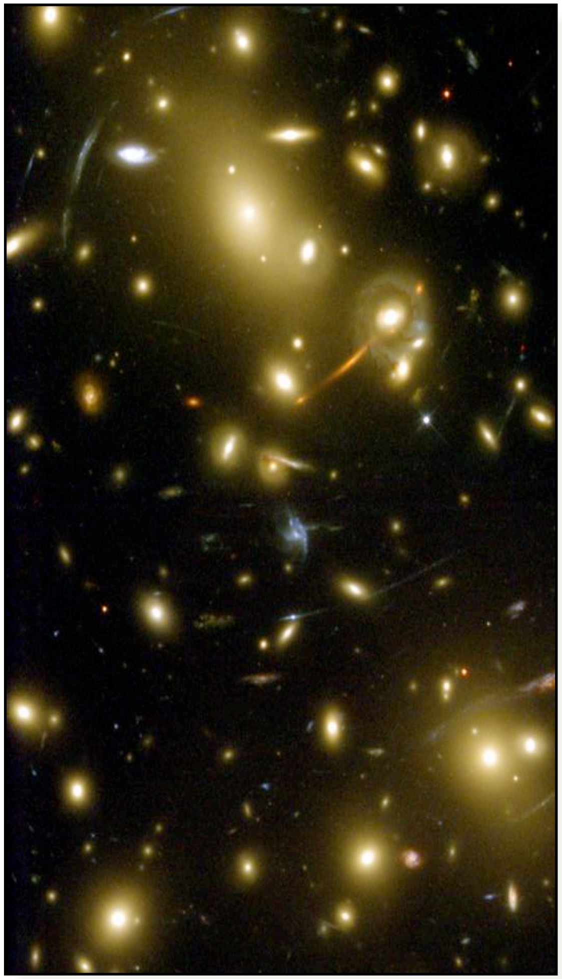
- Goal is to build a computing environment for preservation of particle physics data from BaBar project for next 5-10 years
- BaBar studies high-energy electron-positron collisions produced at the SLAC National Accelerator Facility in Stanford

- 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

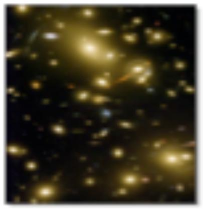
- CANFAR is a partnership between
  - University of Victoria
  - University of British Columbia
  - National Research Council Canadian Astronomy Data Centre
- Will provide computing infrastructure for 6 astronomy survey projects



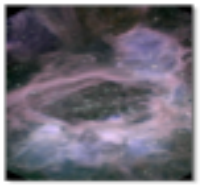
- Each of these surveys requires a different processing environment, which require:
  - A specific version of a Linux distribution
  - A specific compiler version
  - Specific libraries
- Applications have little documentation
- These environments are evolving rapidly

Common Solution:

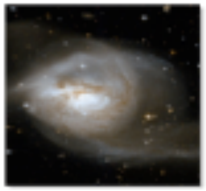
- Create Virtual Machines with these applications installed
- Run jobs for these projects on these VMs
- Users can customize the VMs to suit their specific needs



NGVS VM



PAndAS VM



SASSy VM

BaBar VM

Custom VM

How do we manage jobs and VMs for those jobs on IaaS?

- With IaaS, we can easily create many instances of a VM image
- How do we run jobs on these resources?
- How do we manage these deployed resources?

One solution: One Click Clusters

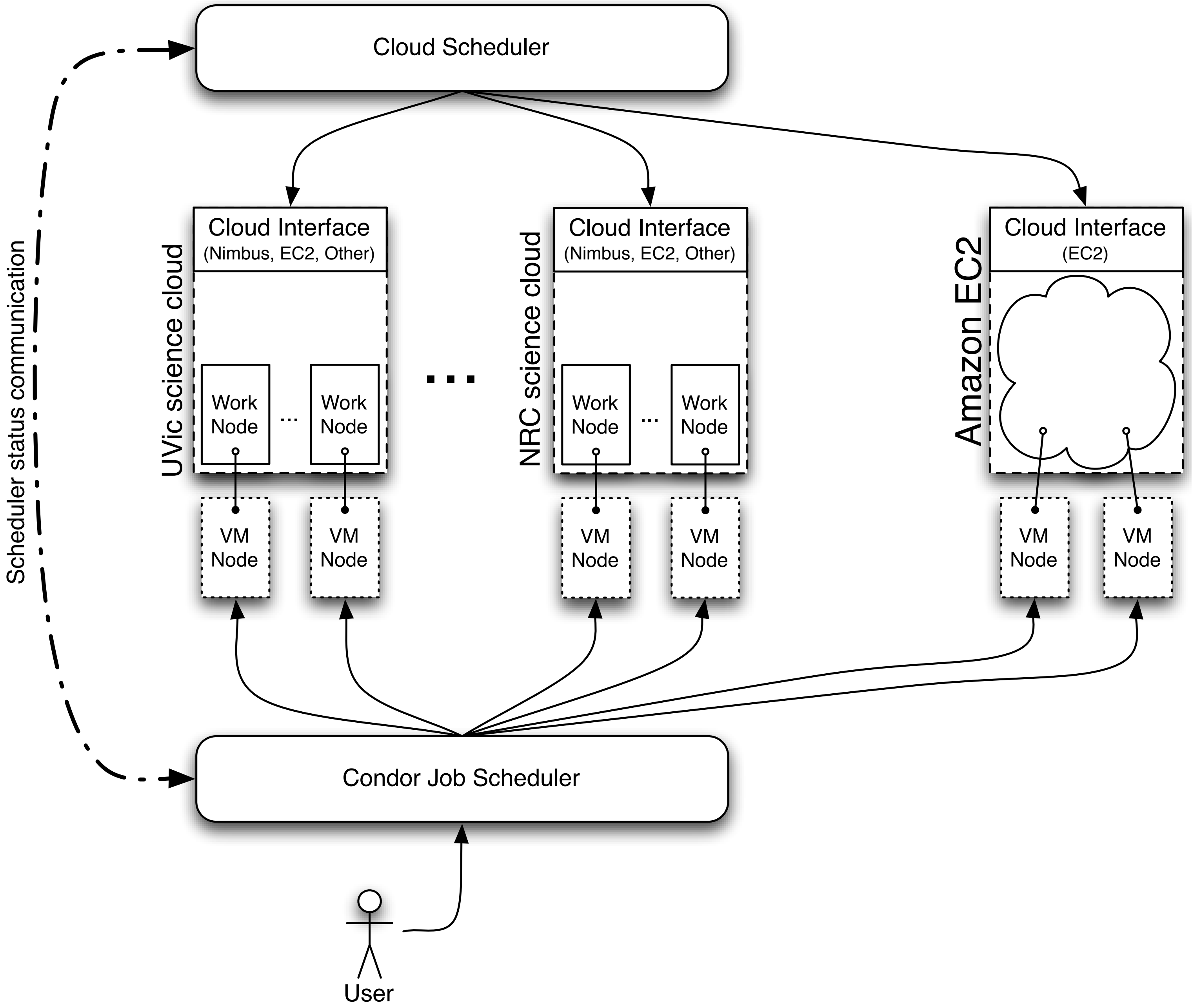
- The Nimbus Context broker allows users to create “One Click Clusters”
- Users create a cluster with their VM, run their jobs, then shut it down
- However, most researchers are used to sending jobs to a HTC cluster, then waiting for those jobs to complete
- Cluster management is unfamiliar to them

Our Solution: Cloud Scheduler

- Users create a VM with their experiment software installed
  - A basic VM is created by our teams, and researchers 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

- Not to compete with existing products
- To be able to use existing IaaS and job scheduler software together, today
- Users should be able to use the HTC tools they are already familiar with
- Adequate scheduling to be useful to our users



Cloud Scheduler
Scheduler status communication
UVic science cloud
Cloud Interface
(Nimbus, EC2, Other)
Work Node
...
Work Node
VM Node
VM Node
...
NRC science cloud
Cloud Interface
(Nimbus, EC2, Other)
Work Node
...
Work Node
VM Node
VM Node
Amazon EC2
Cloud Interface
(EC2)
VM Node
VM Node
Condor Job Scheduler
User

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 Eucalyptus and OpenNebula

Implementation Details

- 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

Preliminary Results

- fill me in with results from Kyle
- Ideally this should show that our solution isn't orders of magnitude slower than One Click Clusters

Future Work

- We are still in the experimental (alpha) stage, so work needs to be done to ensure this can be used in a production environment
- We would like the Cloud Scheduler to consider the carbon footprint of the resources it uses (part of GreenIT)
  - For example, a user could have a carbon budget, and would prefer his jobs to run on sites that produce less carbon

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