An Overview of Cloud Scheduler

Frank Berghaus

University of Victoria

Overview

I. Our Group

II. Cloud Scheduler

III. The ATLAS Grid of Clouds

IV. Belle II & Cloud Scheduler

I. Our Group

- **Randall Sobie**
  - Project Leader
- **Frank Berghaus**
  - Application Specialist
- **Ian Gable**
  - Network Specialist
- **Colin Leavett-Brown**
  - Cloud Developer
- **Michael Paterson**
  - Cloud Developer
- **Ron Demerais**
  - Software Engineer
- **Ryan Taylor**
  - Computing Specialist
- **Andre Charbonneau**
  - Computing Specialist

- Collaborating with
  - Nimbus
  - CERN
  - Caltech, U. Michigan
  - Internet 2
  - ESNet
  - NECTAR

Our Website: http://heprc.phys.uvic.ca/

Our Projects

- *Cloud Scheduler*: Cloud federation software
- *Shoal*: Dynamic squid management
- ATLAS cloud production system
- ATLAS and virtual tier 2 management
- Data intensive applications on distributed clouds
- Software Defined Networks for cloud applications
- Virtual machine management and distribution
- Puppet contextualization of VMs for HEP applications

Overview

I. Introduction to the Group

II. Cloud Scheduler

II. Cloud Scheduler

- Cloud Scheduler is a python package for managing VMs on IaaS clouds
- Users submit HTCondor jobs
  - Optional attributes specify virtual machine properties
- Developed at UVic and NRC since 2009
- Used by ATLAS, CANFAR, and BaBar

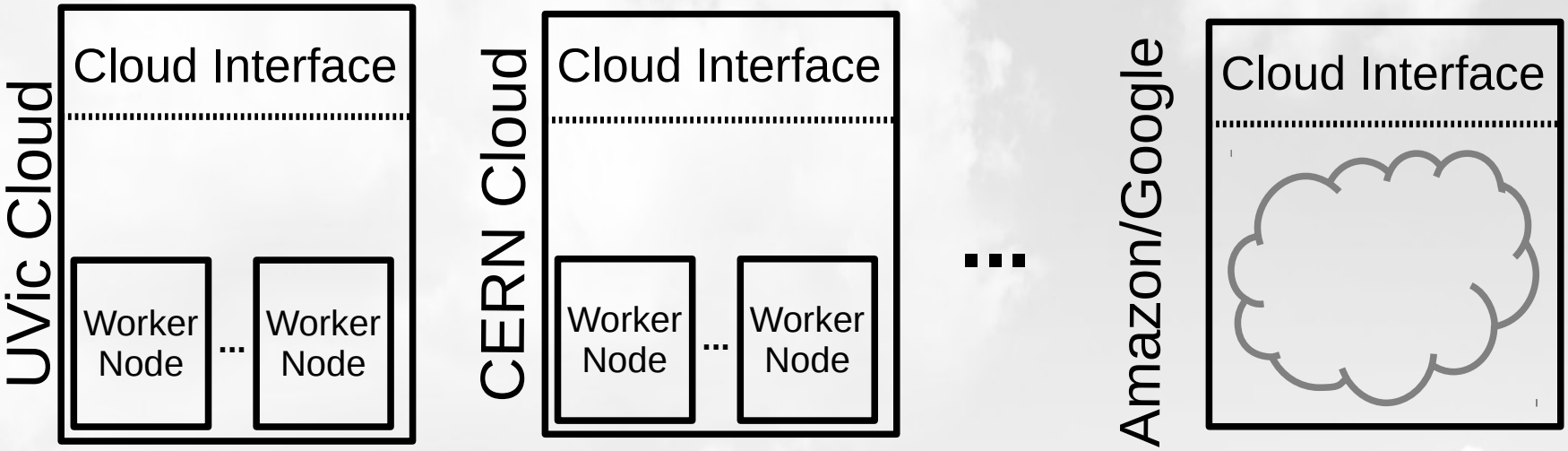
  - The Code: https://github.com/hep-gc/cloud-scheduler
  - Website: http://cloudscheduler.org/
  - Publication: http://arxiv.org/abs/1007.0050

Key Features of Cloud Scheduler

- Dynamically manages quantity and type of VMs in response to user demand
- Easily connects to many IaaS clouds, and aggregates their resources
- Provides IaaS resources in the form of an ordinary HTCondor batch system
- Generic tool, not grid or HEP specific

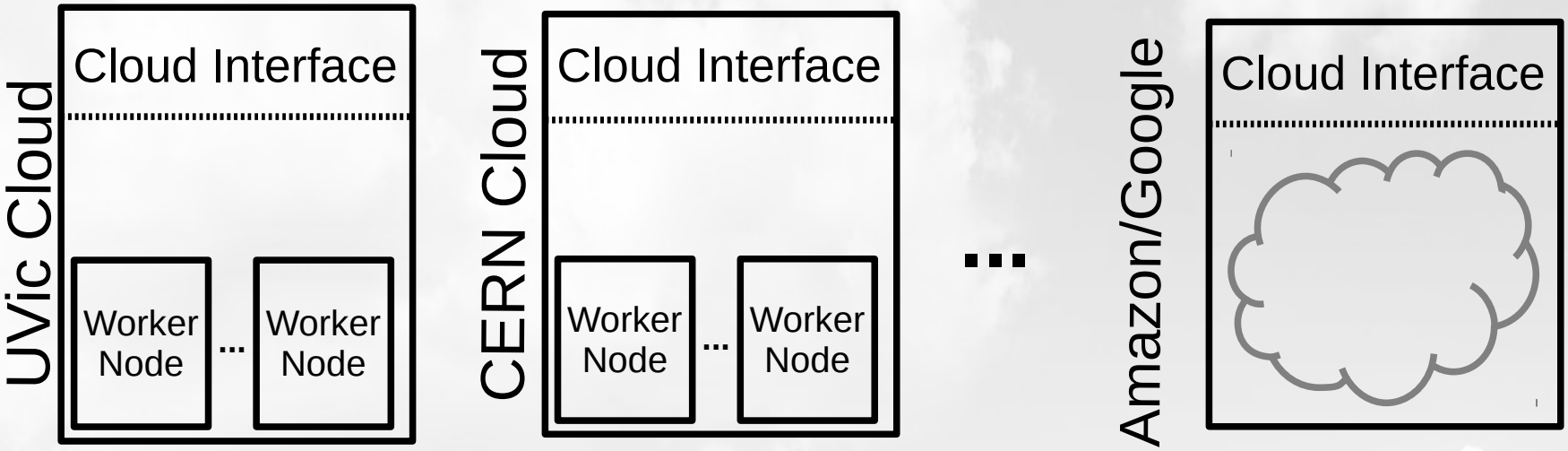
```
pip install cloud-scheduler
```

Step 1

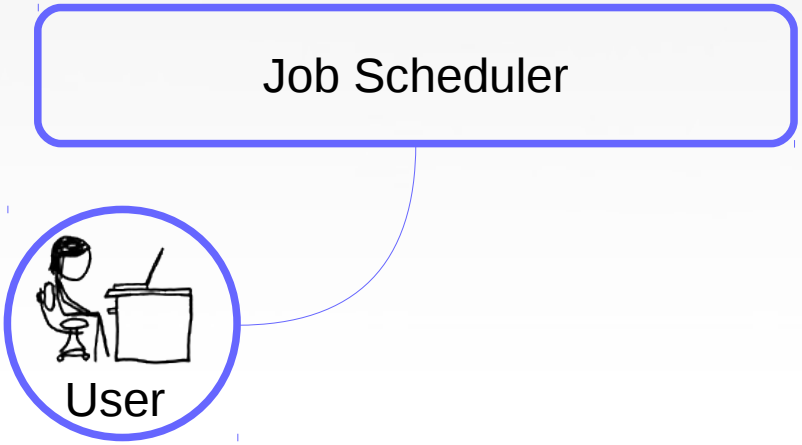


- Supported cloud types:
  - OpenStack
  - Nimbus
  - StratusLab
  - OpenNebula
  - Amazon EC2
  - Google Compute Engine
- Research and commercial clouds made available through a cloud interface

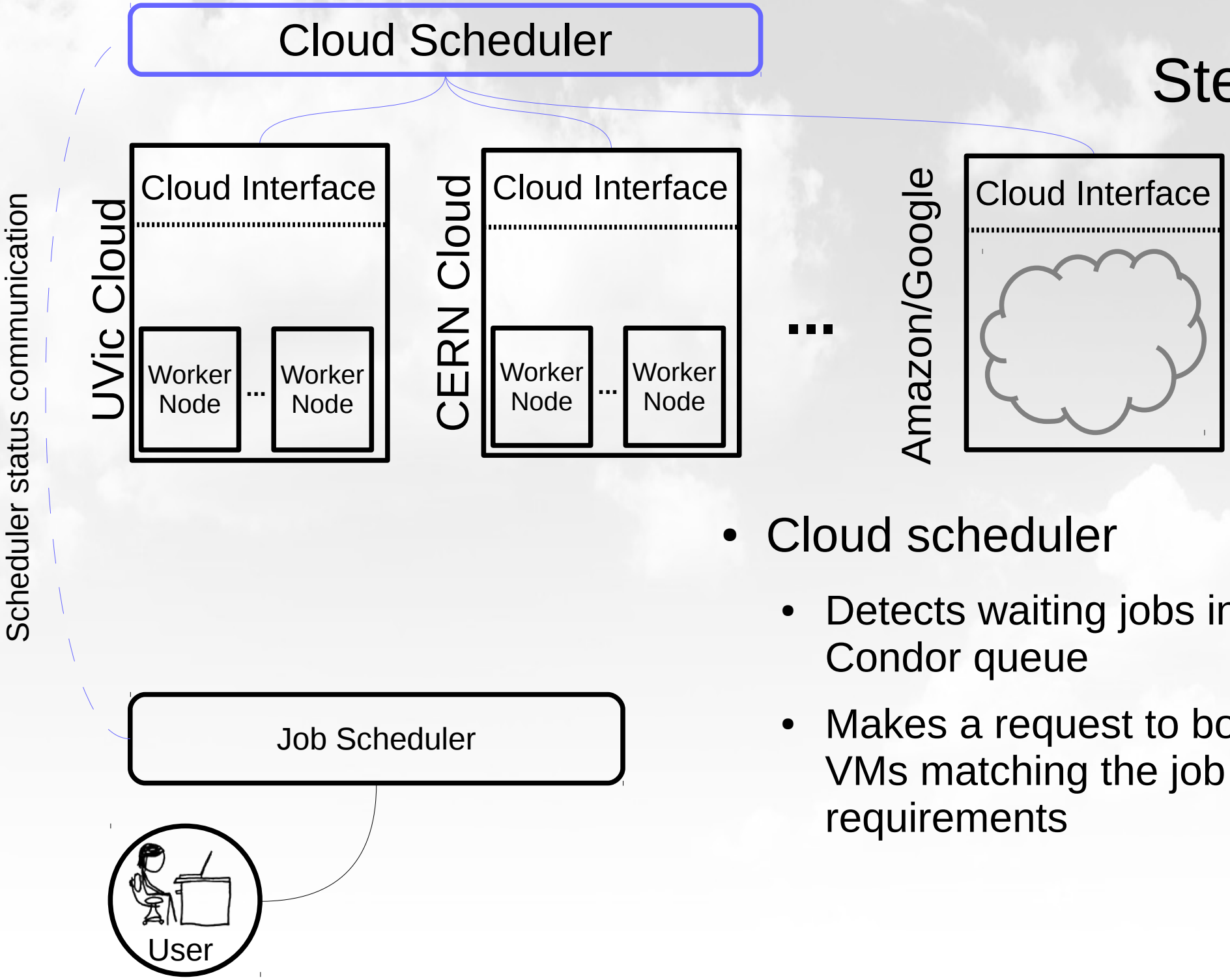
Step 2



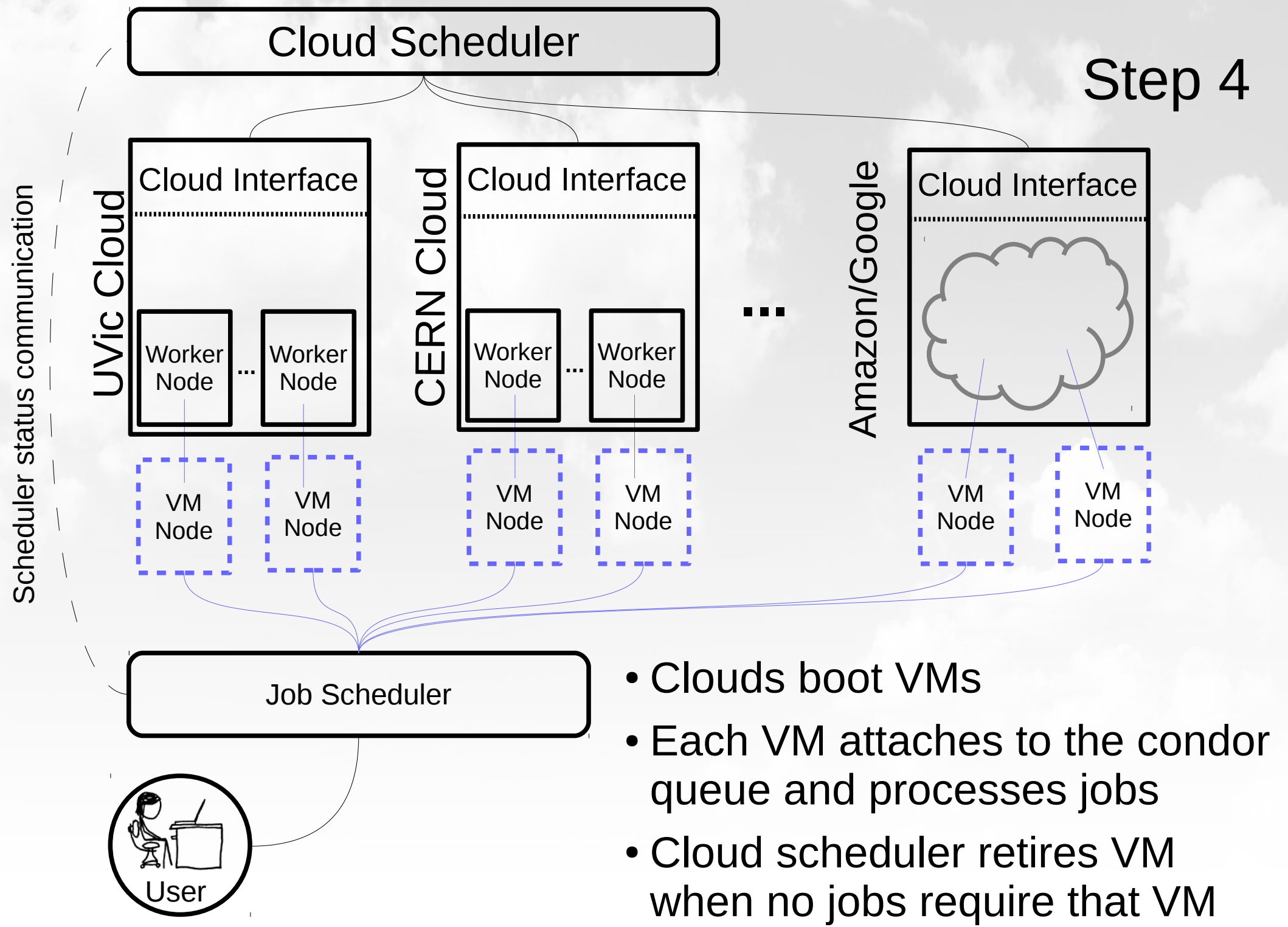
- User submits job to HTCondor
- Job scheduler may not have any resources yet



Step 3



- Cloud scheduler
  - Detects waiting jobs in Condor queue
  - Makes a request to boot VMs matching the job requirements



- Clouds boot VMs
- Each VM attaches to the condor queue and processes jobs
- Cloud scheduler retires VM when no jobs require that VM

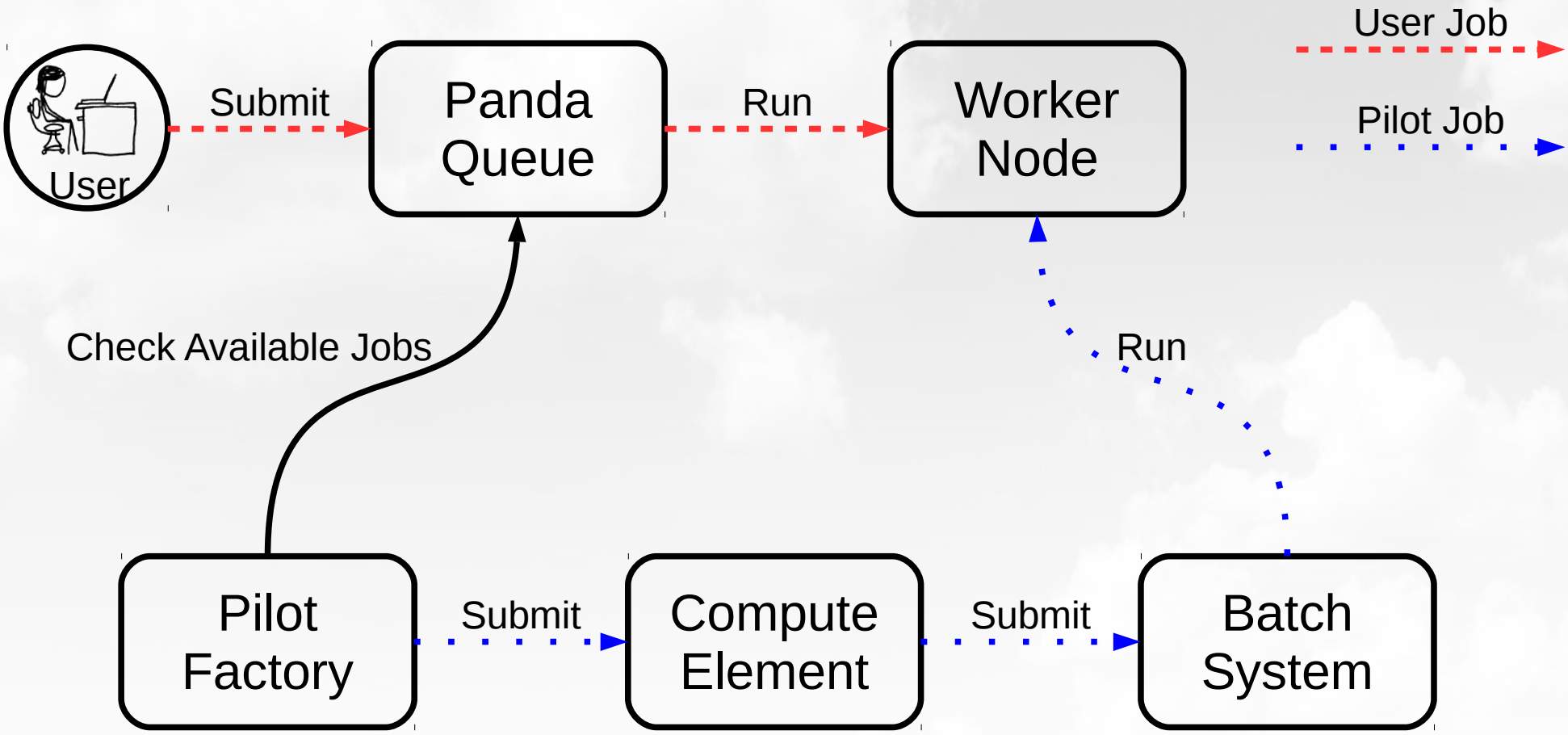
Overview

I. Introduction to the Group

II. Cloud Scheduler

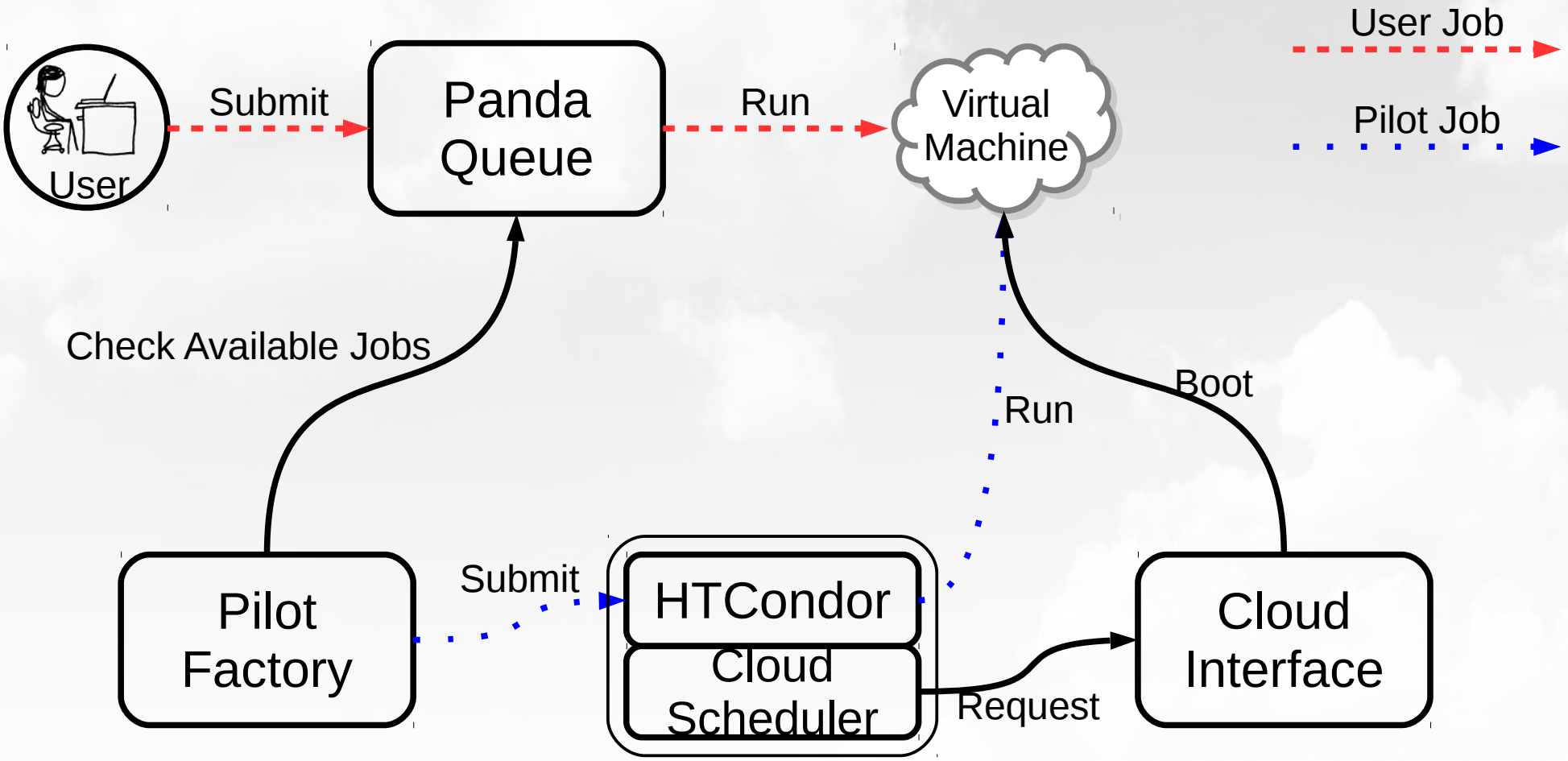
III. The ATLAS Grid of Clouds

Grid Job Flow



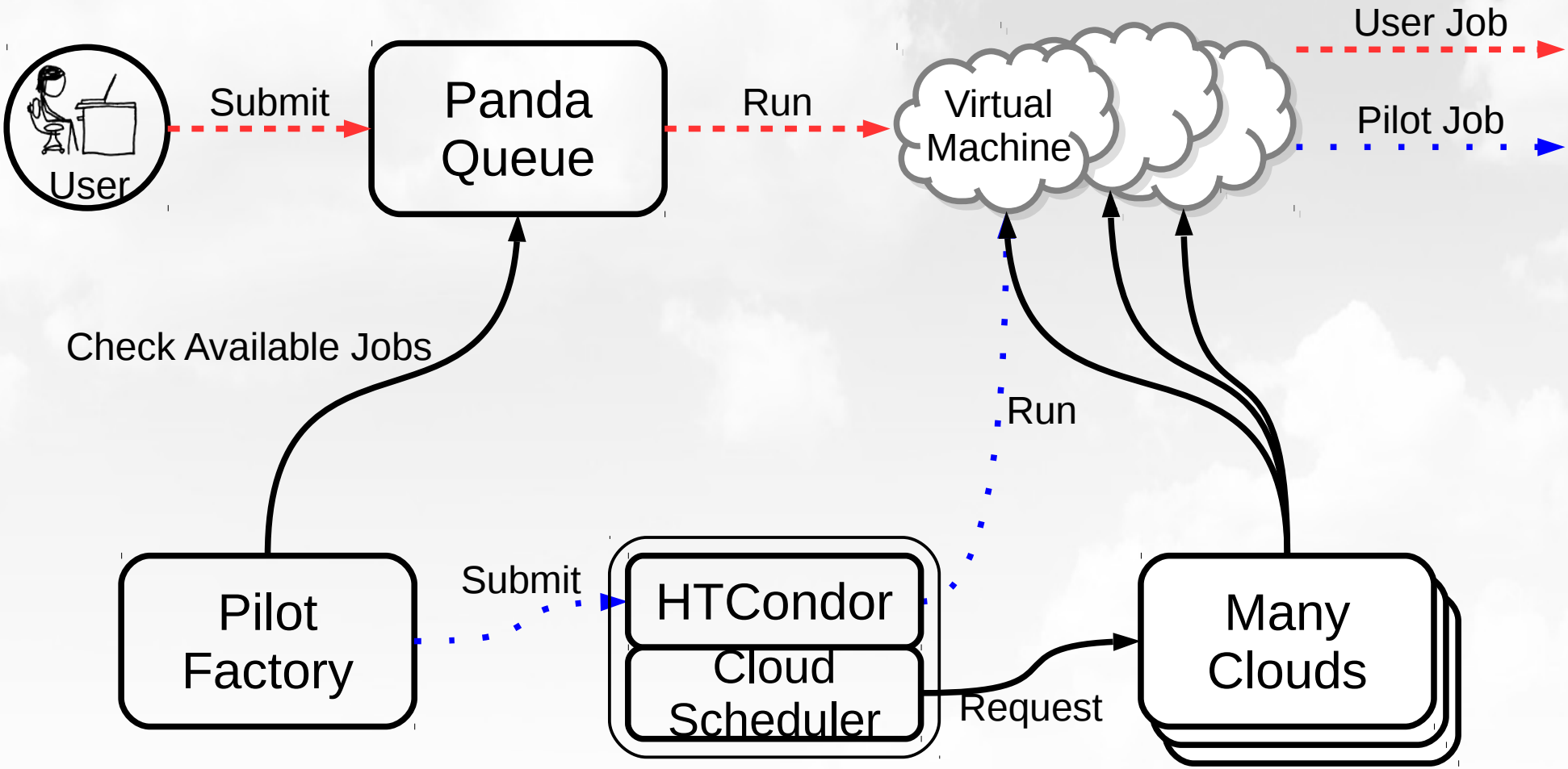
- Compute Element is tightly coupled to batch system

Cloud Job Flow (on the Grid)



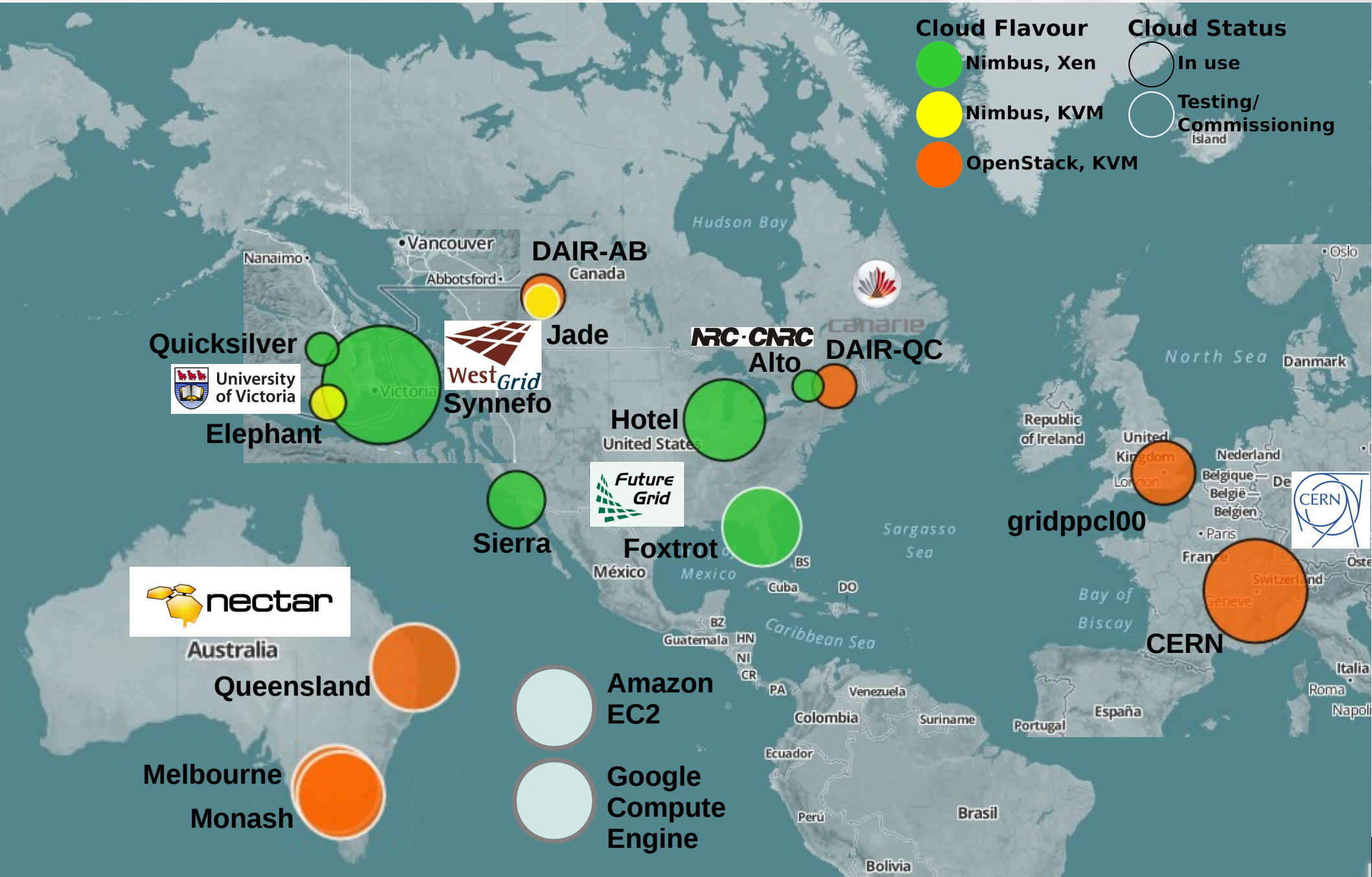
- Cloud Scheduler is loosely coupled to cloud interface

Cloud Job Flow (on the Grid)



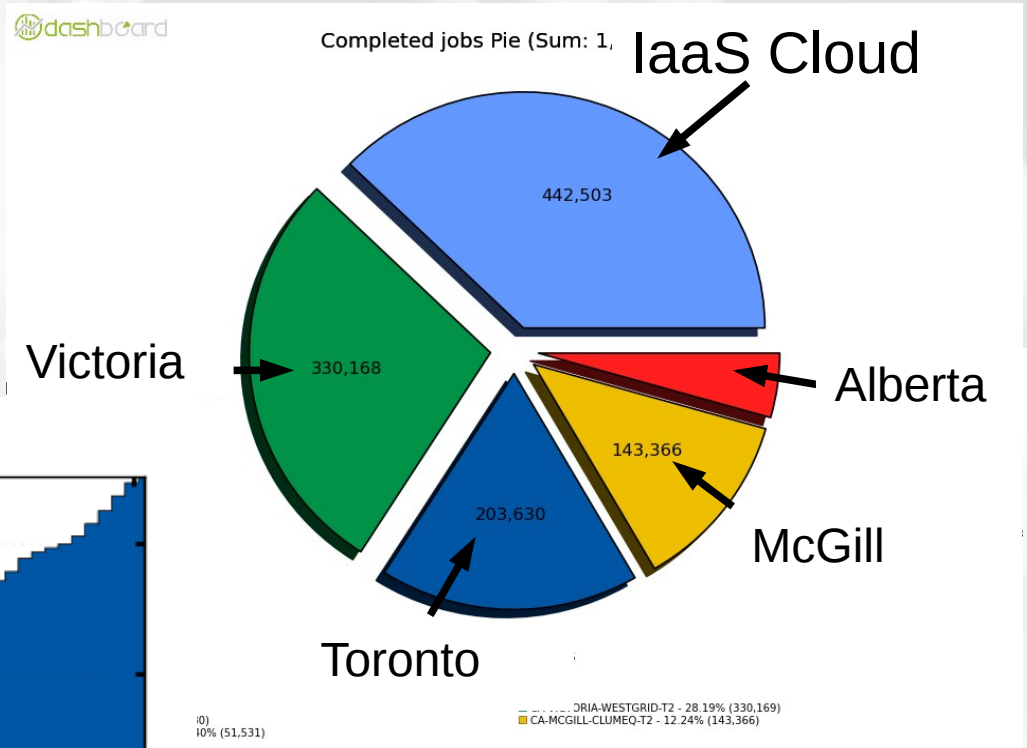
- Easy to connect and use many clouds

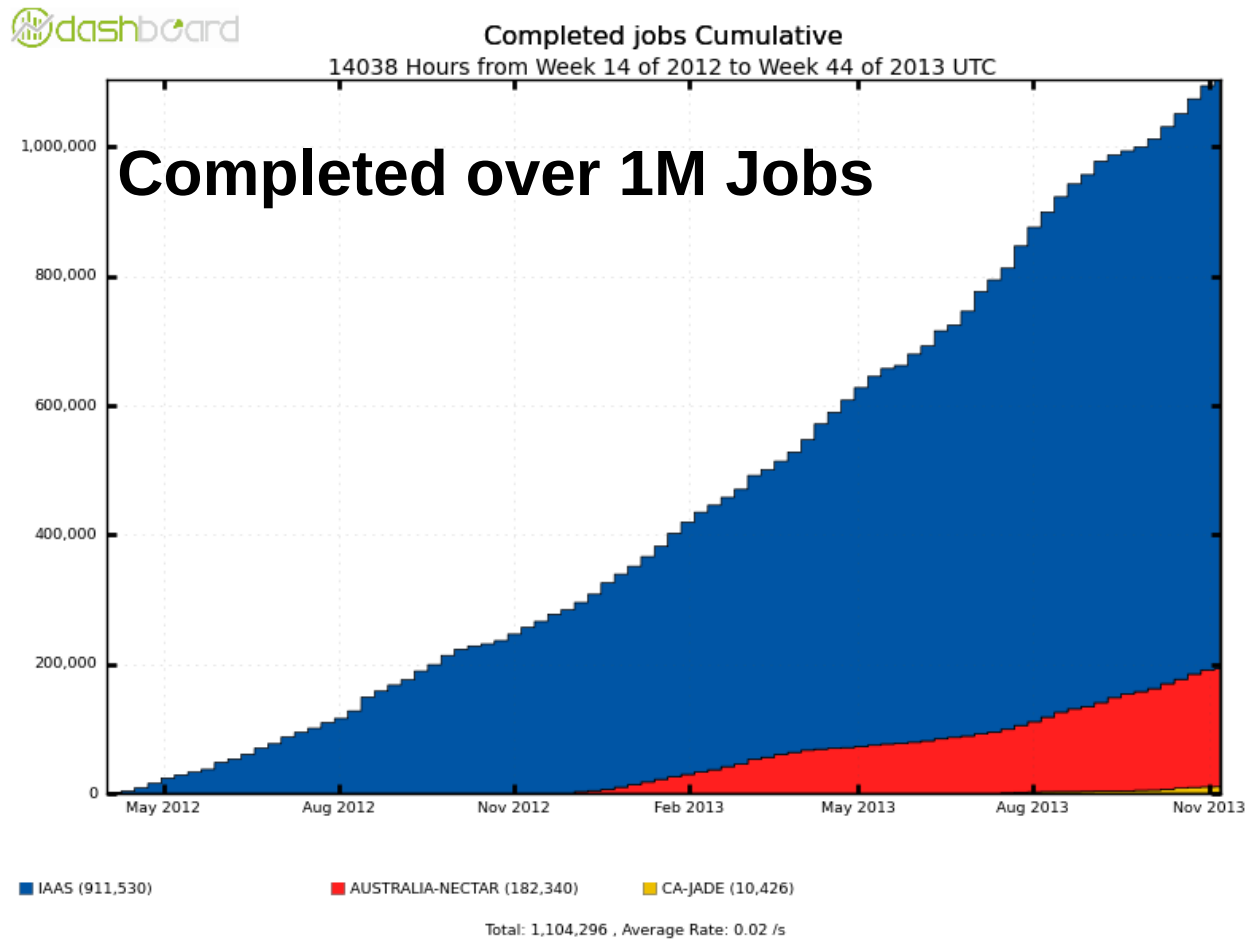
The ATLAS “Grid of Clouds”



Cloud Production in ATLAS

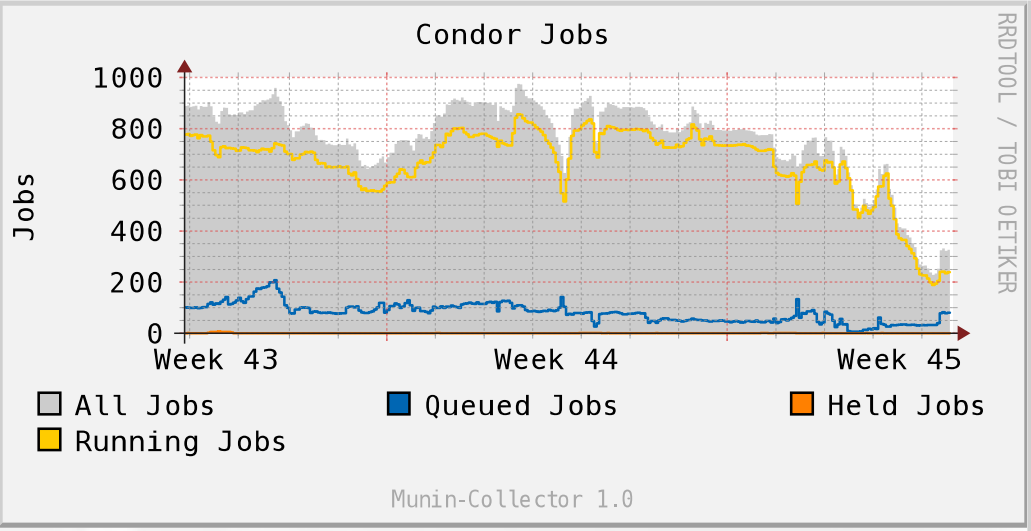
- Started operation April 2012

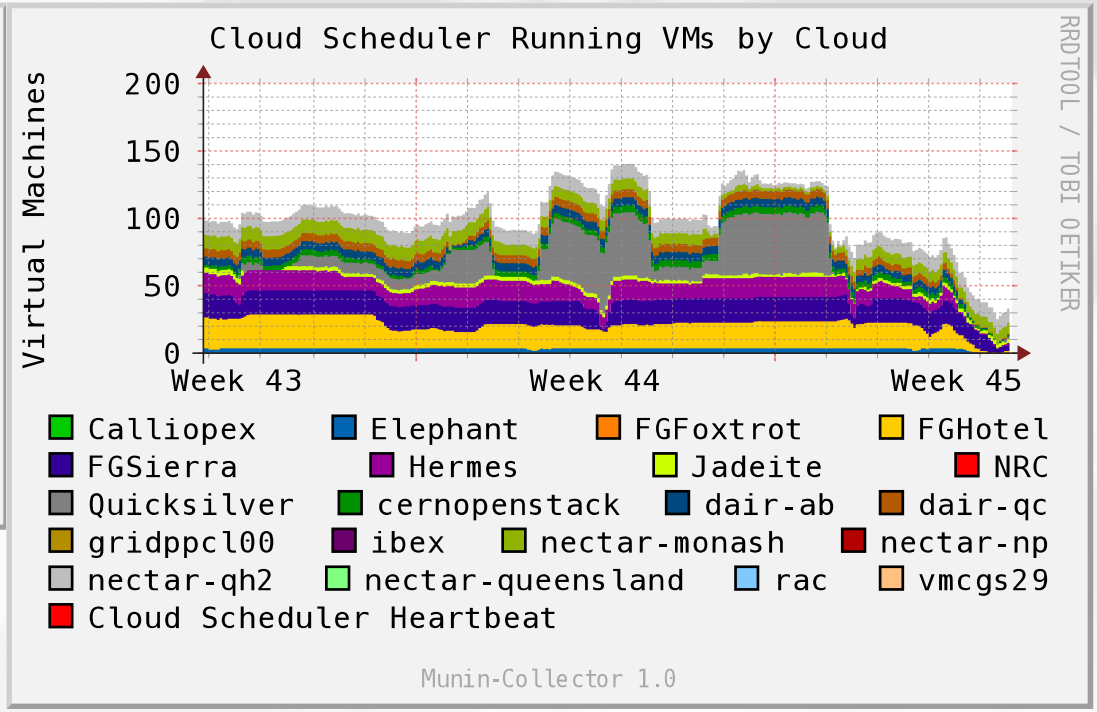




- Similar performance to dedicated facilities at
  - University of Victoria
  - McGill University
  - University of Alberta
  - University of Toronto

ATLAS Cloud Workload





- ATLAS jobs and virtual machines over the last two weeks
- VMs retiring in response to job queue

- Virtual machines with 8 cores
- VMs on Nimbus clouds have 1 week lifetime

Overview

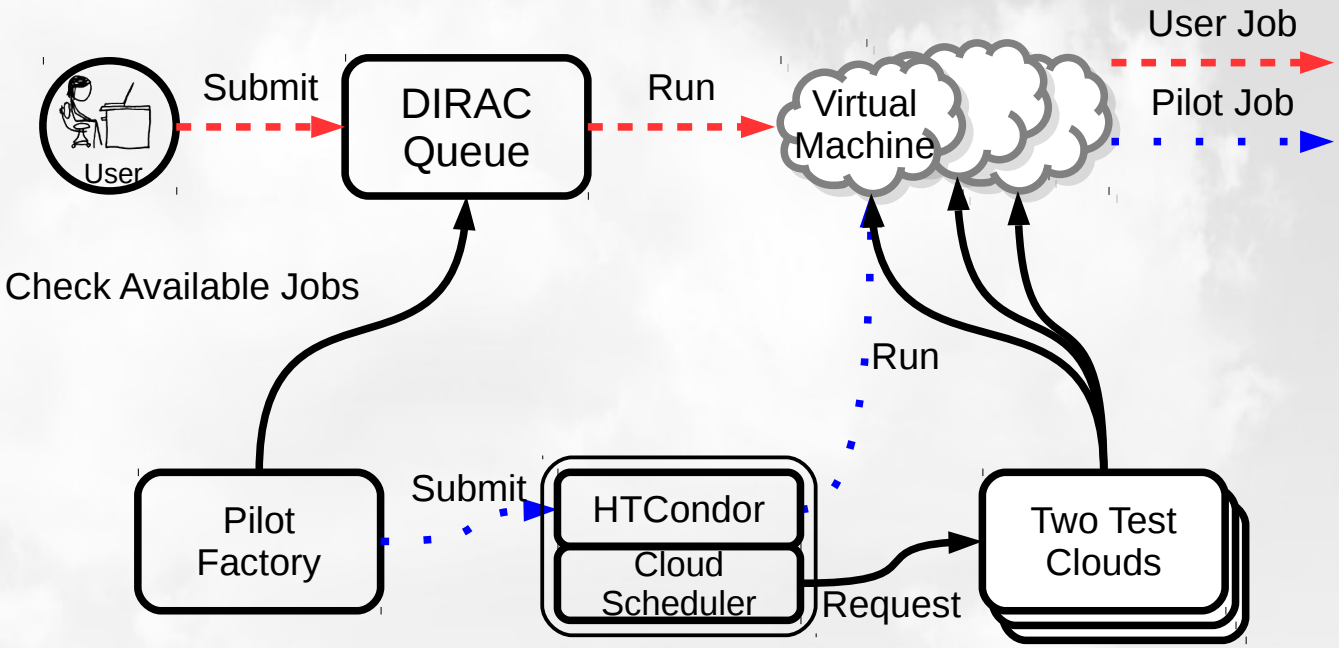
I. Introduction to the Group

II. Cloud Scheduler

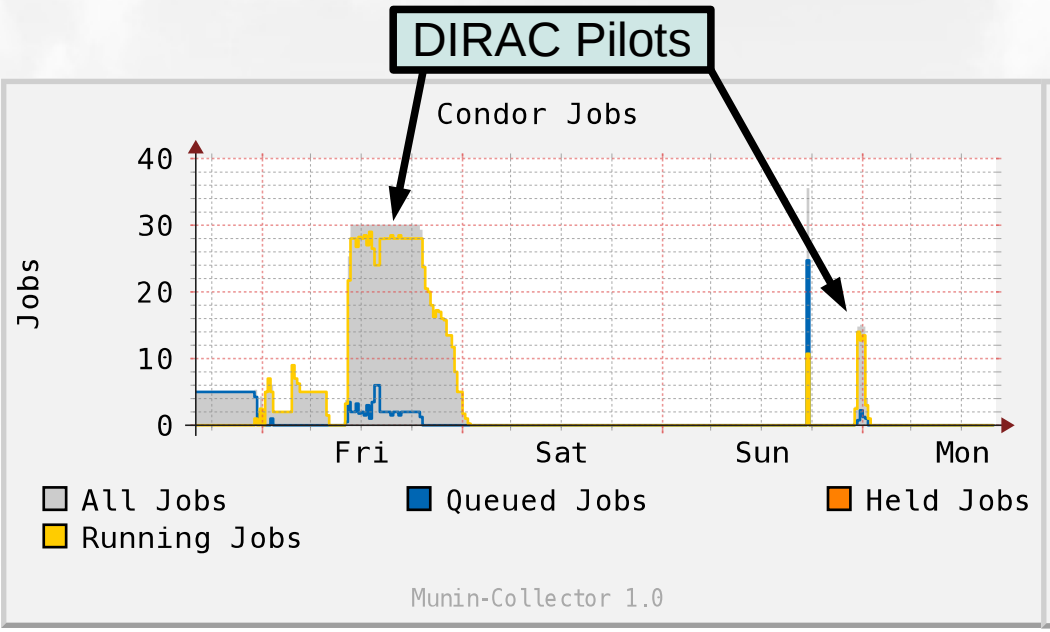
III. The ATLAS Grid of Clouds

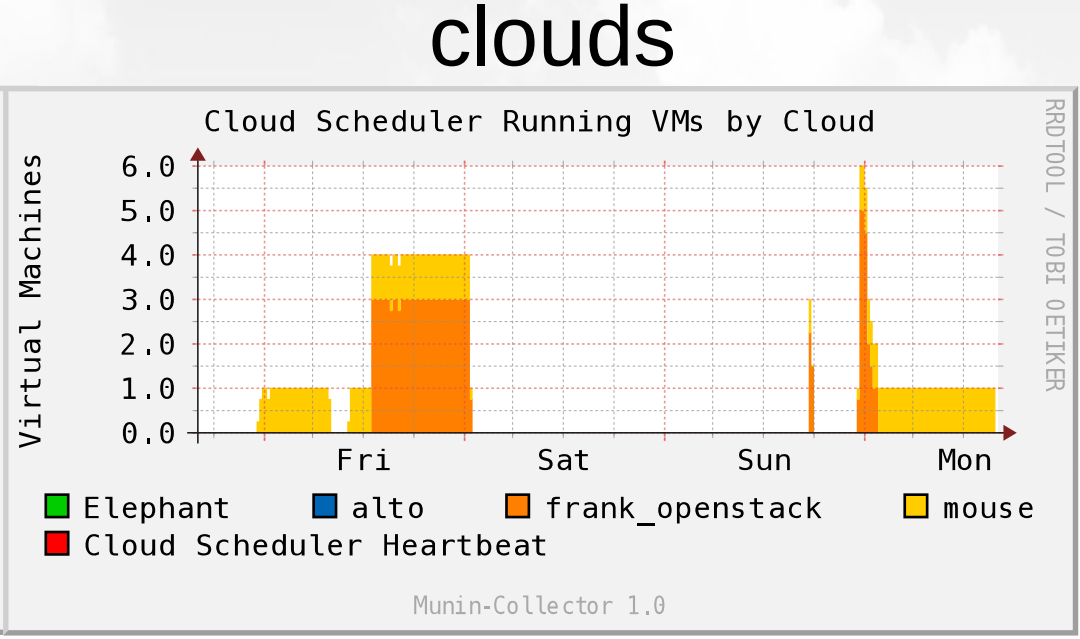
IV. Belle II & Cloud Scheduler

Belle II & Cloud Scheduler



- Integrated into DIRAC
- Running Belle jobs since November 8
- Using two test clouds





- Thanks to Miyake-san and Malachi!

Belle II VM Images

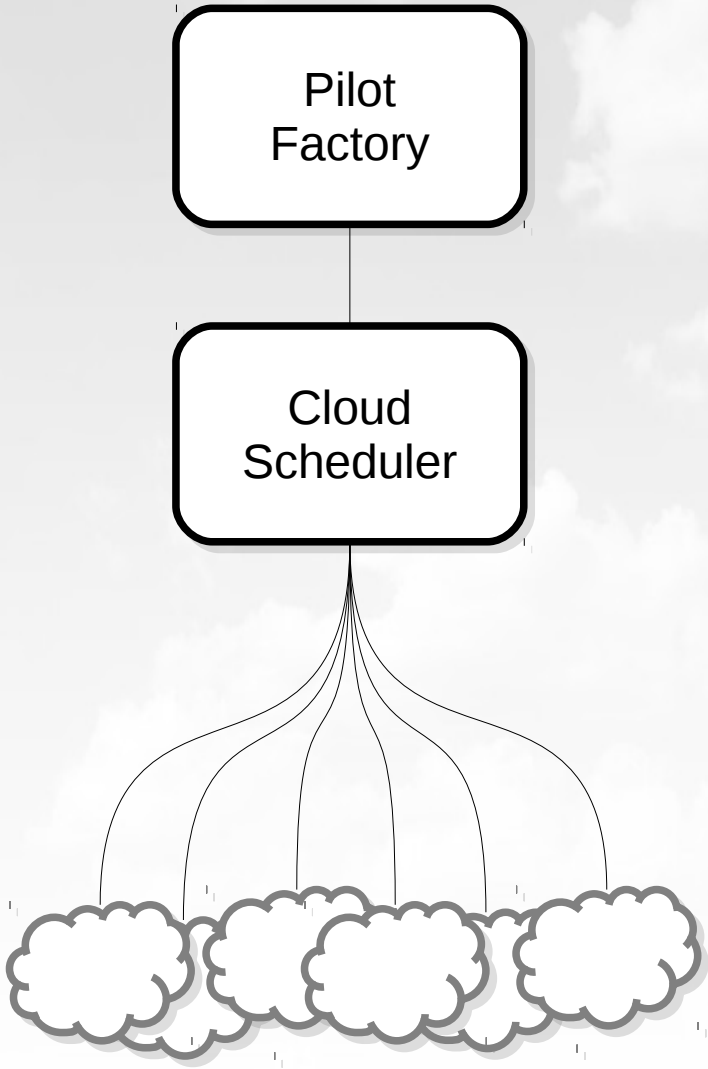
- Based on CernVM 2.7.2 (SL5)
- Belle II software is made available over cvmfs
- Puppet contextualizes images on boot

  https://github.com/MadMalcolm/atlasgce-modules

  - Developed for ATLAS, Belle II required minor modifications
- Same image can work on any
  - Hypervisor (xen or kvm)
  - Cloud Type (OpenStack, GCE, Nimbus, EC2, etc.)
  - Cloud Location

Advantages to Belle II

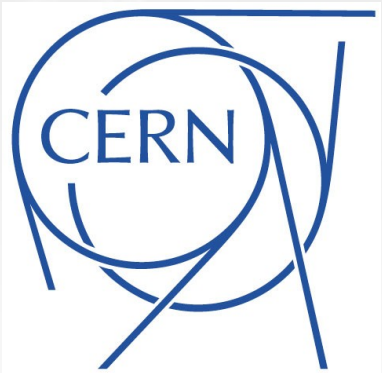
- Layer above the resources
- Access many resource sites, using few Cloud Scheduler servers
- No Belle-specific configuration or services needed at resource site
- Opportunistic integration of cloud resources



Cloud Scheduler Summary

- Federate HEP and non-HEP academic resources in single queue
- Share resources with other projects
- Used in production by ATLAS and CANFAR for over 1.5 years
- Successfully integrated with DIRAC
- Access to non-HEP funded cloud development group
- Limited only by resource availability
- Complimentary to VMDIRAC

Acknowledgements



Backup

Connecting Additional Clouds

- Add a few lines to a configuration file
  - /etc/cloudscheduler/cloud_resources.conf

```
[MyCloud]
host: mycloud.example.org
cloud_type: OpenStack
vm_slots: 50
networks: private
enabled: true
```

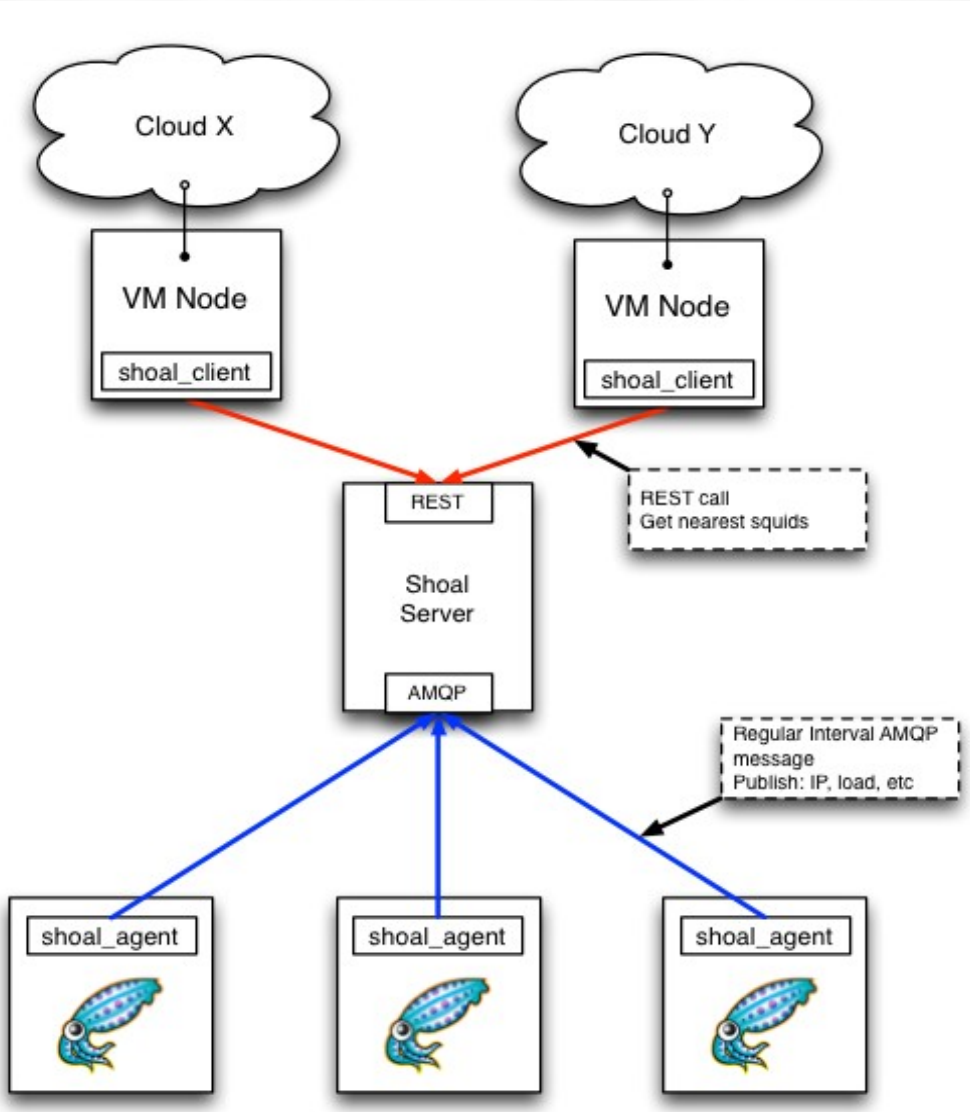
- Get authorization on the cloud
  - Secret key or x509 proxy
- Test booting virtual machines

Optional Job Description

```
# Run-environment requirements
Requirements = VMType =?= "cernvm-belle-node-2.7.2-x86_64" && \
               Arch == "x86_64"


# User requirements
+VMName          = "cernvm-batch-node-2.7.2-x86_64"
+VMCPUArch       = "x86_64"
+VMAMI           = "mouse01.heprc.uvic.ca:ami-00000070"
+VMInstanceType = "mouse01.heprc.uvic.ca:m1.large"
+VMMem           = "8192"
+VMCPUCores      = "4"
```

Shoal: Dynamic Squid Management



- Need robust network of squids – especially with µCernVM
- Boot squid VMs in each cloud on demand
- VMs automatically discover and use local squids
- CHEP 2013 Poster
- WLCG HTTP Proxy Discovery Task Force
- Code:

  https://github.com/hep-gc/shoal
- Paper:

  http://arxiv.org/abs/1311.0058

HTCondor

- Designed as cycle scavenger
- Ideal as job scheduler in a dynamic environment
- Job description specifies VM image and requirement:

- Nimbus requires:
  - VM image URL
  - User proxy

- OpenStack Requires
  - VM image AMI
  - Instance Type

- Cloud scheduler supports default settings