

Distributed cloud computing

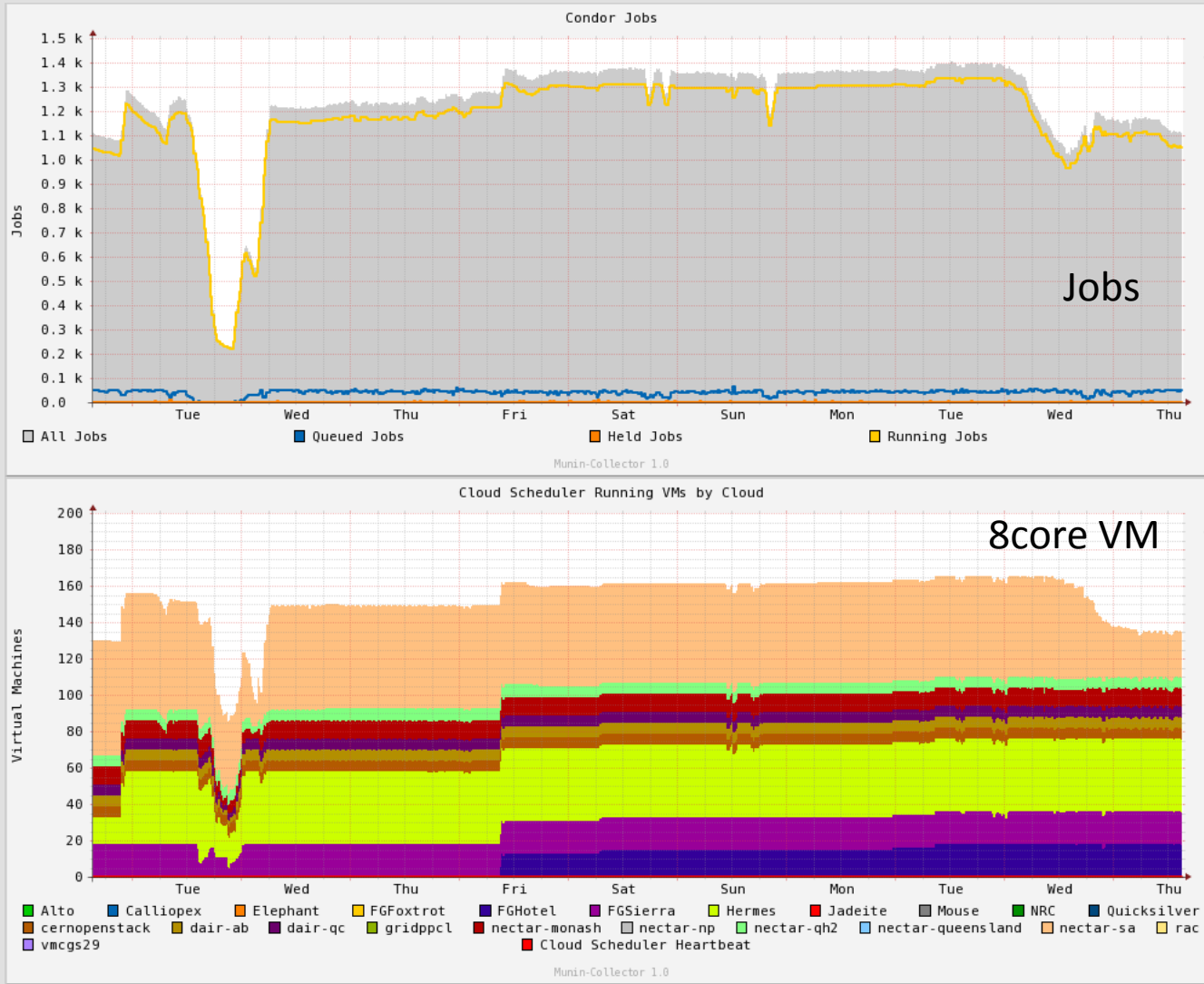
R.Sobie, I.Gable, R.Taylor, M.Paterson, F.Berghaus, C.Leavett-Brown and R.Desmarais
University of Victoria

- We operate a distributed cloud system for ATLAS production jobs
 - In operation since April 2012 (3 continents, many IaaS clouds)
 - Primarily using opportunistic resources
 - Comparable in size to a Canadian ATLAS Tier 2 site
 - See the overview presentation at July 2013 pre-GDB meeting
- Integrated into Belle-II system (using DIRAC)
- Also used for CANFAR Project (Canadian astronomy computing)

Collaborative Activities

- VM management with Puppet
 - Preparing tests with Google Compute Engine (GCE)
- Dynamic software caching (Squids/Shoal)
 - *Shoal* discovers, tracks and advertises dynamic *Squids*
- Installation of *CloudScheduler* at CERN using Wigner cloud
- Analysis jobs on the cloud
 - Using webdav/FAX to retrieve the data from the nearest SE
- ATLAS Multicore jobs
- Cloud bursting on EC2 and GCE (EC2 grant)

Dynamic resources (easy to use idle cycles over the holiday period)



Discussion points I

- Clouds as a dynamic computing resources
 - Simplify management of dedicated resources
 - Access to opportunistic resources
 - All resources, services, data storage become dynamic
- Resource sharing
 - Resources are allocated statically (typically limited to N cores per cloud)
 - Condor supports multi-users with a sharing algorithm
 - CloudScheduler will retire VMs of one user if another user-job has priority
 - This can take some time (12 hours) to gracefully retire a multi-core VM running ATLAS
 - Our VMs run until there is no jobs requiring it (we run many jobs on a single VM)
 - Alternative model would be to run 1-job in 1-VM (and let the local scheduler deal with job priority)
 - Future will be easier with multi-core jobs

Discussion points II

- Security
 - Cloud APIs: OpenStack (access key/secret key), Nimbus (x509)
 - The VMs are contextualized at boot time with credentials that allow connection to condor pool
 - Once VM is booted, pilots are pushed to the VM and credentials are managed like any other WLCG job
 - VMs image easily auditable or recreated - built from base-VM distro using Puppet.
 - Logging and security traceability is non-existent. Images are destroyed at the end of life.
- Accounting
 - Using opportunistic resources, so it has not been an issue
 - Techno-political task