CMS Experience Provisioning Cloud Resources with GlideinWMS

Anthony Tiradani

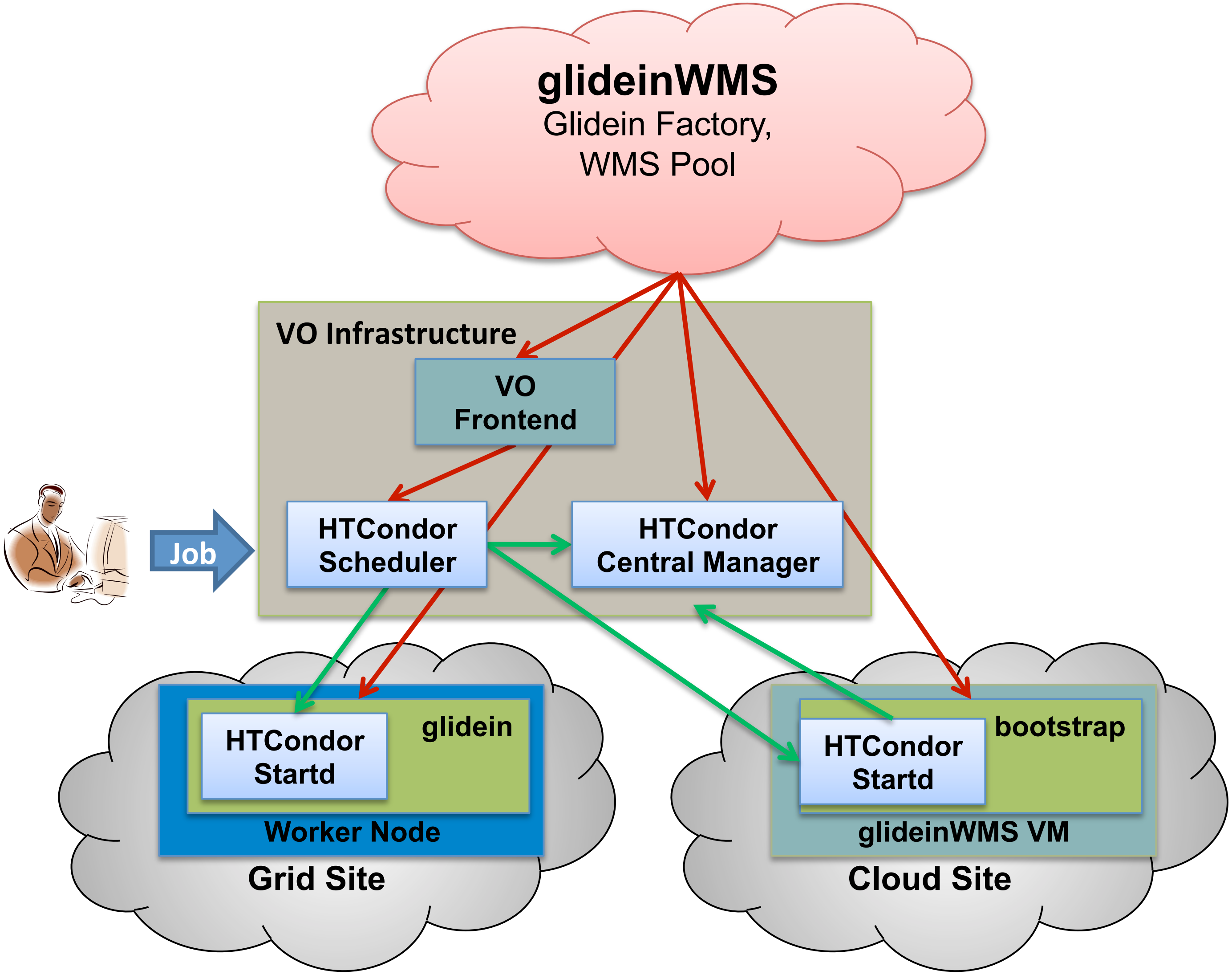
HTCondor Week 2015

20 May 2015

glideinWMS Quick Facts

- glideinWMS is an open-source Fermilab Computing Sector product driven by CMS
- Heavy reliance on HTCondor from UW Madison and we work closely with them
- http://tinyurl.com/glideinWMS

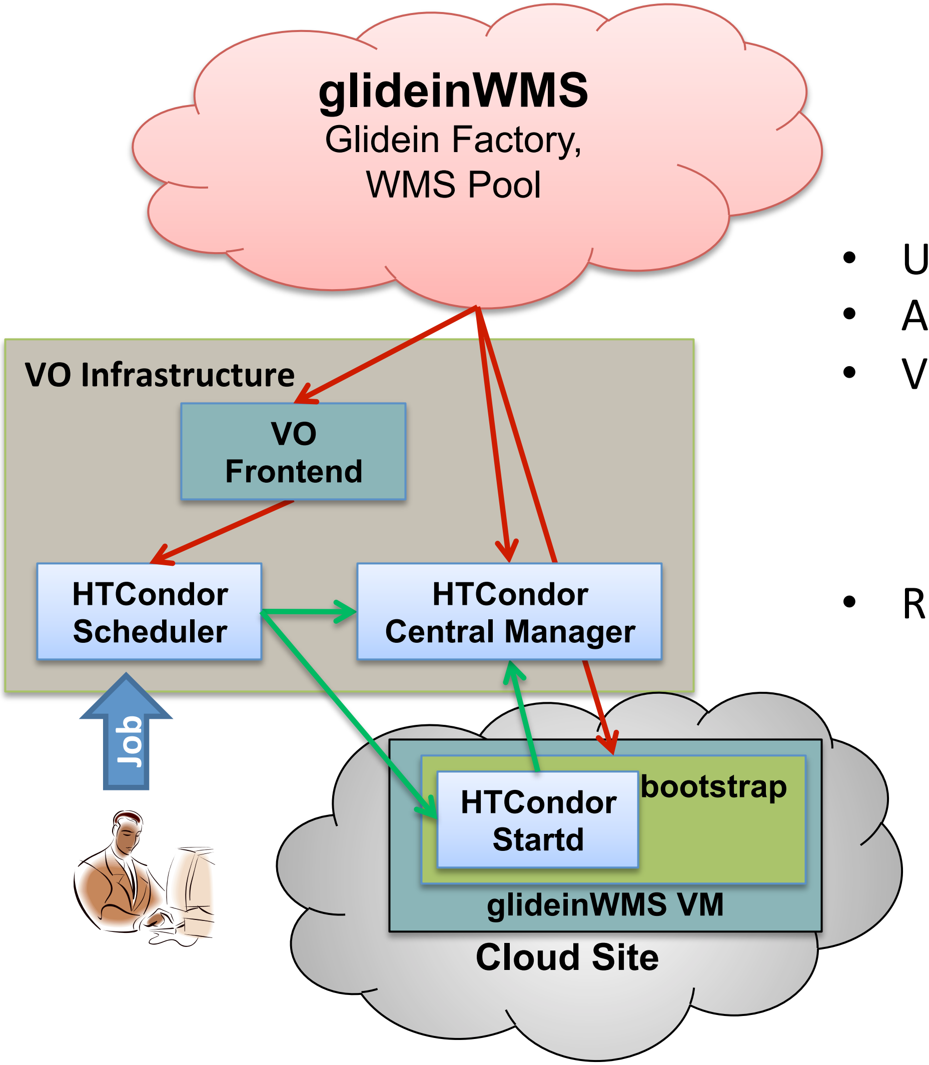
GlideinWMS Overview



CMS Use of CERN AI

- CERN AI: CERN Agile Infrasructure
- T0: (Tier-0) CERN Data Center traditionally where first pass data processing takes place (See Dave Mason’s talk for details)
- T0 completely moved from LSF to AI (OpenStack)
- Approximately 9000 cores now (-> 15000=~220 kHS06)
- 8 core/ 16GB VMs
- Images built by CMS, using CERN-IT automated build system.
- VM are not very dynamic but almost static (1 month duration)
- Resources are part of the T0 Pool and shared with other uses. However, T0 runs with very high priority and so will push out other users quickly if needed.

GlideinWMS & CERN AI



- Use The EC2 API
- AI had stability issues, mostly fixed now
- VMs have ~1 month lifetime
  - Still suffer from long ramp up times
  - Due to resource scarcity, danger of losing resources if they are released
- Results have been mostly favorable

CMS HLT as a Cloud

- HLT: High Level Trigger farm
- The HLT is a considerable CPU resource. At startup:
  - Dell C6100 : 288 nodes, type = 12 cores 24GB ram (will be decommissioned end of 2015 and replaced)
  - Dell C6220 : 256 nodes, type = 16 cores 32GB ram (will be decommissioned end of 2016 and replaced)
  - Megware : 360 nodes, type =24 cores 64GB ram (are the brand new machines)
- No usable disk mass storage (only local disk to each machine)
- 60Gb/s network connection to CERN IT
- When needed to be the HLT it must be the HLT and alternative use must not interfere
  - However when not the HLT this is an extremely valuable resource. Cloud solution chosen (based on OpenStack) with VMs that can be killed completely if needed.

CMS HLT as a Cloud

- Even with expected overheads we can expect typically ~6 hours of usable time between runs. But this time is randomly distributed.
- So anticipate running ~2hour jobs, but this will be adjusted as we gain experience.
- Even during fills use of the HLT is not constant so should be possible to sometimes use part of the HLT even during data taking.

CMS HLT as a Cloud

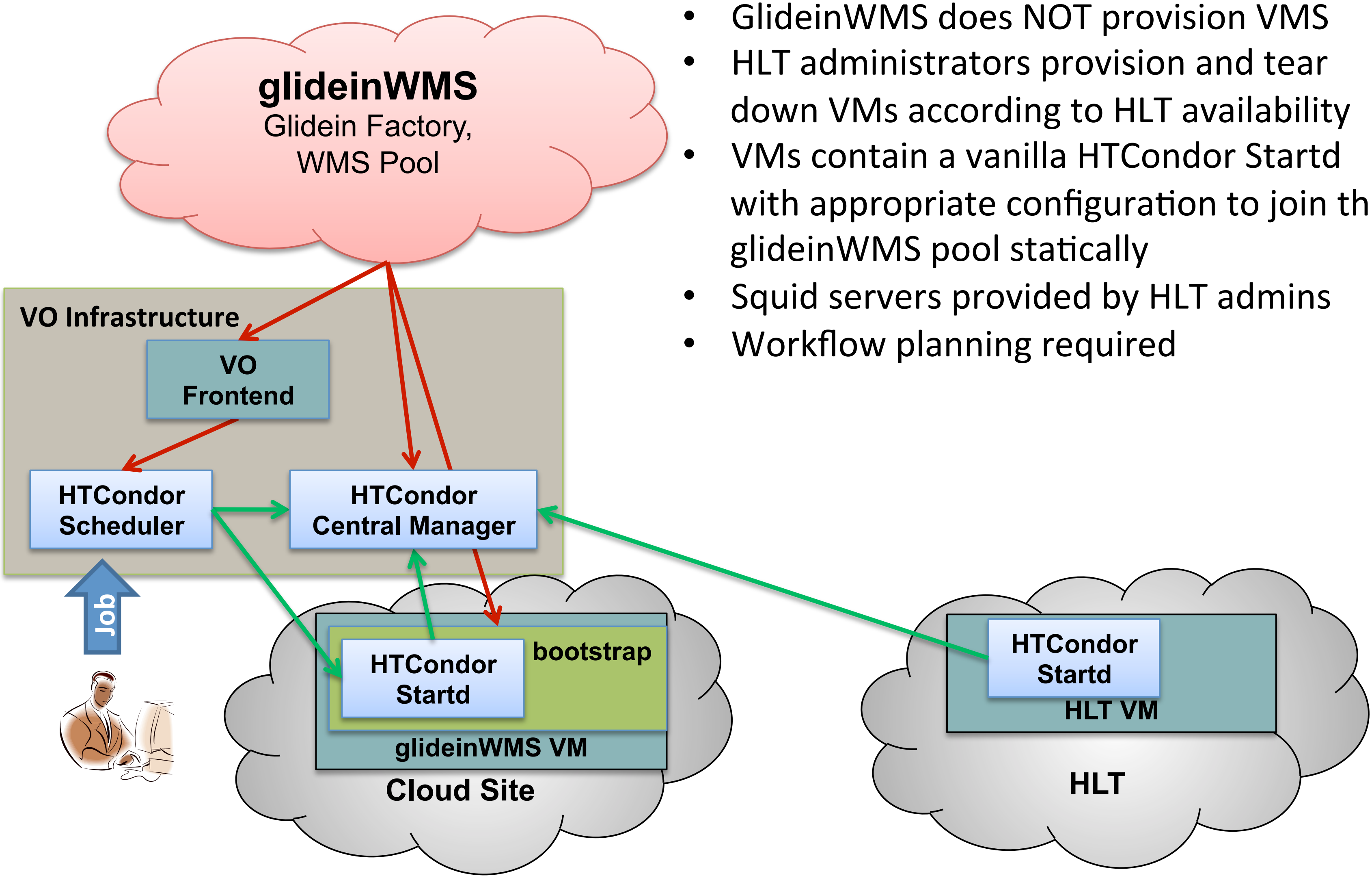
- So we have a 3 level plan:

- Start by using HLT as a Cloud during Technical stops. If that works …
- Start to use the HLT as a Cloud between fills. If that works …
- Start to use parts of the HLT as a Cloud during fills at low luminosity.

- However, in this model the HLT team must be in control.

- So a new tool, Cloud-igniter, used by the online team to start and kill VMs rather than requests from coming from the factory. New Vms connect to the Global pool.

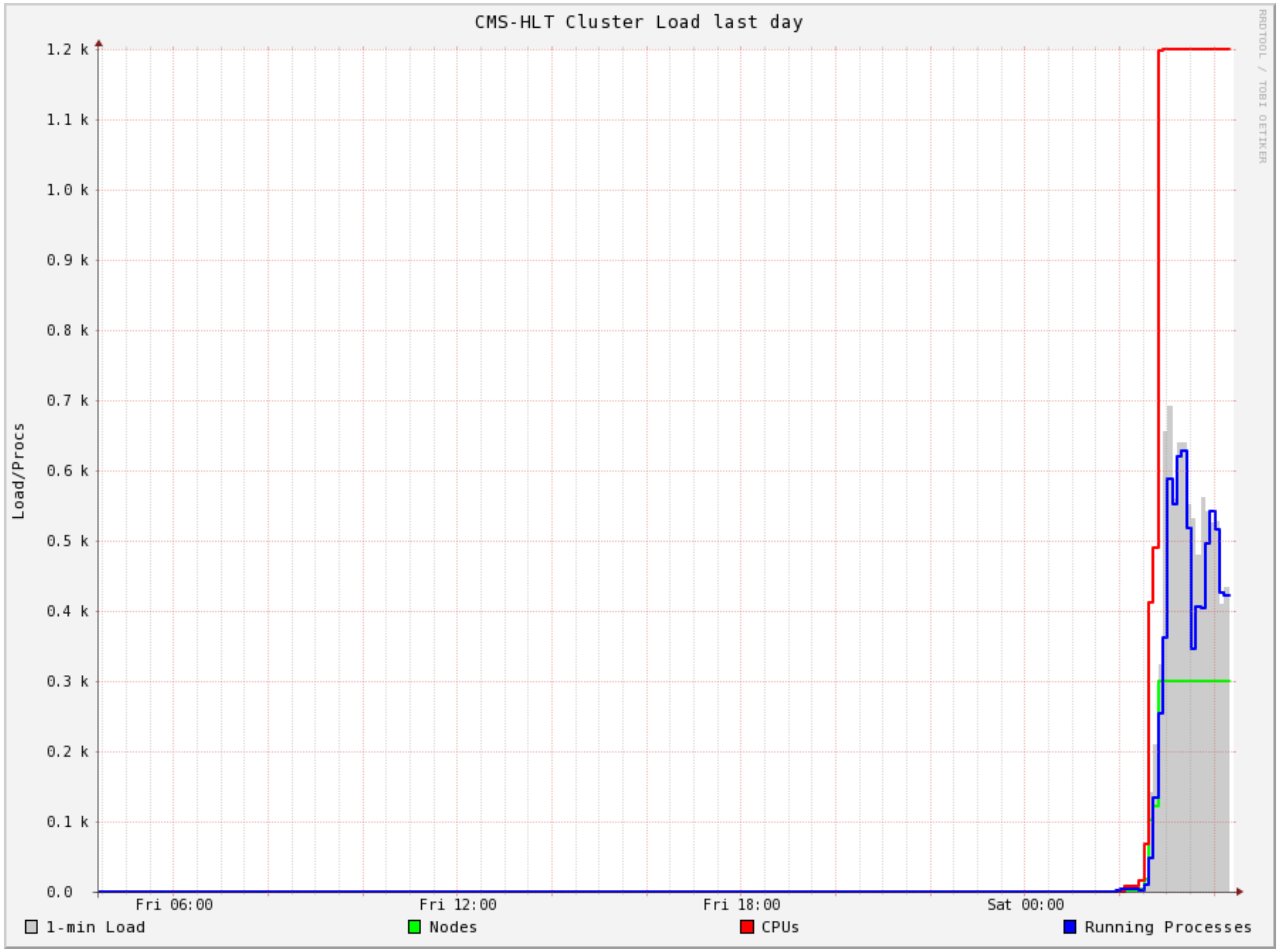
GlideinWMS & CMS HLT



- GlideinWMS does NOT provision VMS
- HLT administrators provision and tear down VMs according to HLT availability
- VMs contain a vanilla HTCondor Startd with appropriate configuration to join the glideinWMS pool statically
- Squid servers provided by HLT admins
- Workflow planning required

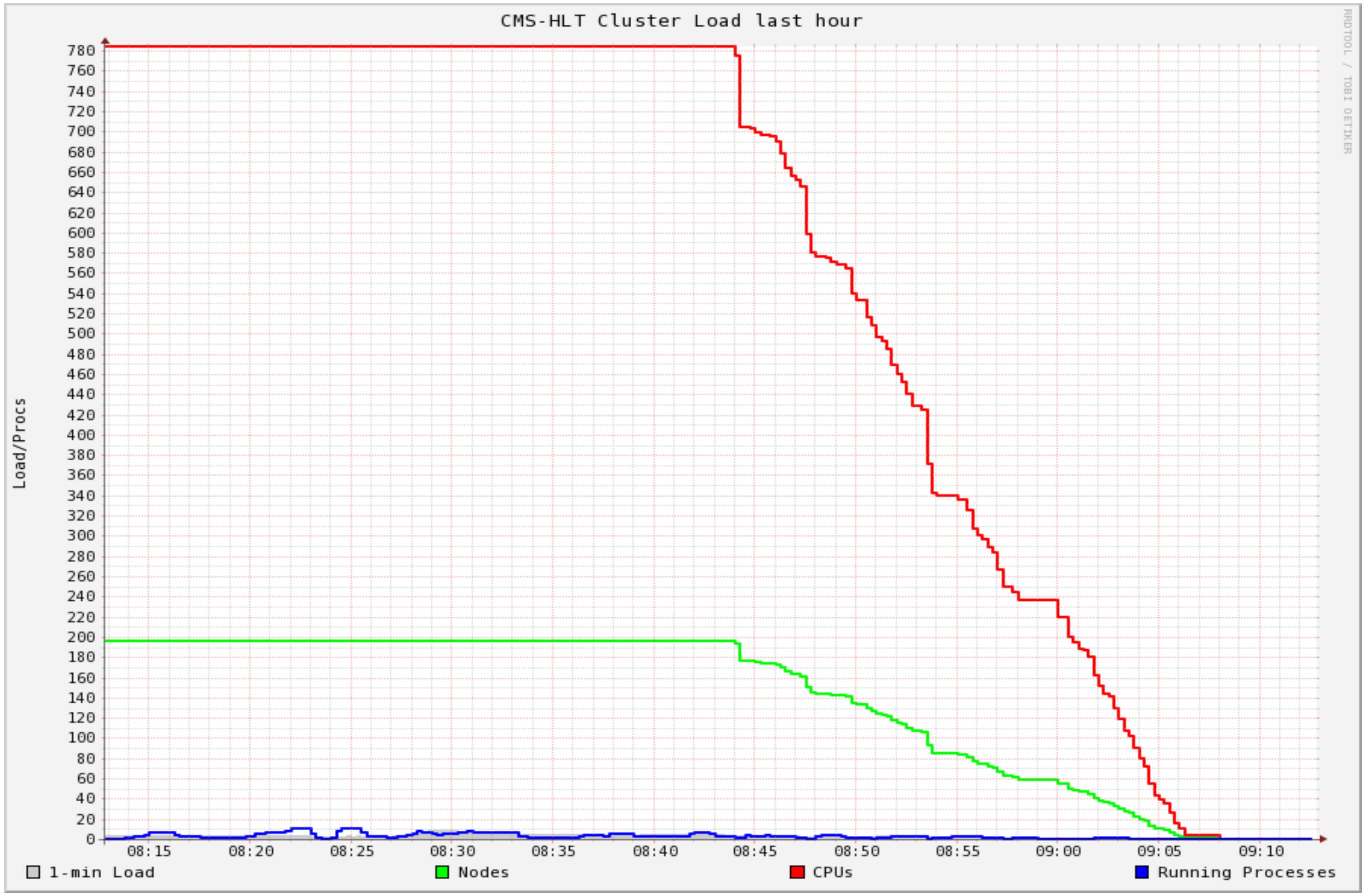
CMS HLT as a Cloud

- Can start 1200 cores in ~10 minutes with EC2 API
- EC2 API faster than NOVA API



CMS HLT as a Cloud

- Can shutdown 700 cores in ~20 minutes with EC2 API – in reality, HLT admins hard kill VMs.
- Takes ~1 minute for entire cluster



Institutional Clouds

- Have been working with clouds in China, Italy and the UK. All OpenStack.
- Have run user analysis using GlideinWMS installed in Italy and the UK.
- Have mainly tested Institutional Clouds using the “traditional” CMS cloud approach. However, have carried tests using other tools

Fermilab OneFacility Project

- The goal is to transparently incorporate all resources available to Fermilab into one coherent facility
- Experiments should be able to perform the full spectrum of computing work on resources regardless of the "location" of the resources.
- Include commercial and community cloud resources
- Include "allocation based" resources via BOSCO

Fermilab OneFacility Project

- Initial work will focus on several use cases:
  - Nova use case tests sustained data processing in the cloud over the course of several months
  - CMS use case tests bursting into the cloud using many resources for a short period
  - Other use cases will explore specialized hardware needs

Cloud Bursting

- Pilot project is to “burst” into AWS
- Want to use ~56K cores (m3.2xlarge) for a month (8 vCPUs, 30GiB RAM, 160GB SSD)
- Initial phase will be provisioned through the FNAL Tier-1 as part of the OneFacility project
- See Sanjay Padhi’s talk for more details
- Would like to acknowledge Michael Ernst and ATLAS for initial work with AWS and sharing lessons learned

Cloud Challenges

- Grid sites provide certain infrastructure
- Cloud sites provide the tools to build infrastructure
- Data placement
- Authentication/Authorization
- VM image management
- Cost management

HTCondor Wishlist

- Make network a first class citizen
  - Network bandwidth management one of the challenges facing the OneFacility project for all use cases
  - Counters for how much data is being transferred in and out, knobs to restrict traffic, etc.
- Native cloud protocols
  - Using EC2 API now
  - Native nova support for OpenStack (EC2 API moving out of the core and into third party development/support)
  - Google Compute Engine support
  - Microsoft Azure support?

HTCondor Wishlist

- Better integration with cloud tooling
  - EC2 API good if a longer ramp-up is tolerable
  - Not so good if you need to burst now
  - Need to build grid infrastructure in the cloud, would be nice to be able to “orchestrate” the architecture on demand (OpenStack Heat, AWS CloudFormation)
- Provide a better view of the total cloud usage orchestrated by HTCondor
  - Classads that contain (as reported by the cloud infrstructure):
    - total computing time
    - total storage usage
    - Total inbound and outbound network usage
    - Usage statistics from other services offered by the infrastructures
- LogStash filters for all HTCondor logs

Acknowledgements

- David Colling (Imperial College) for use of CHEP talk
- CMS support: Anastasios Andronidis, Daniela Bauer, Olivier Chaze, David Colling, Marc Dobson, Maria Girone, Claudio Grandi, Adam Huffman, Dirk Hufnagel, Farrukh Aftab Khan, Andrew Lahiff, Alison McCrae, Massimo Sgaravatto, Duncan Rand, Xiaomei Zhang + support from many other people
- ATLAS: Michael Ernst and others for sharing knowledge and experiences
- glideinWMS team
- HTCondor Team

Questions?