CMS Cloud Work

Mattia Cinquilli
IT-SDC-OL

IT-SDC-OL Cloud Team

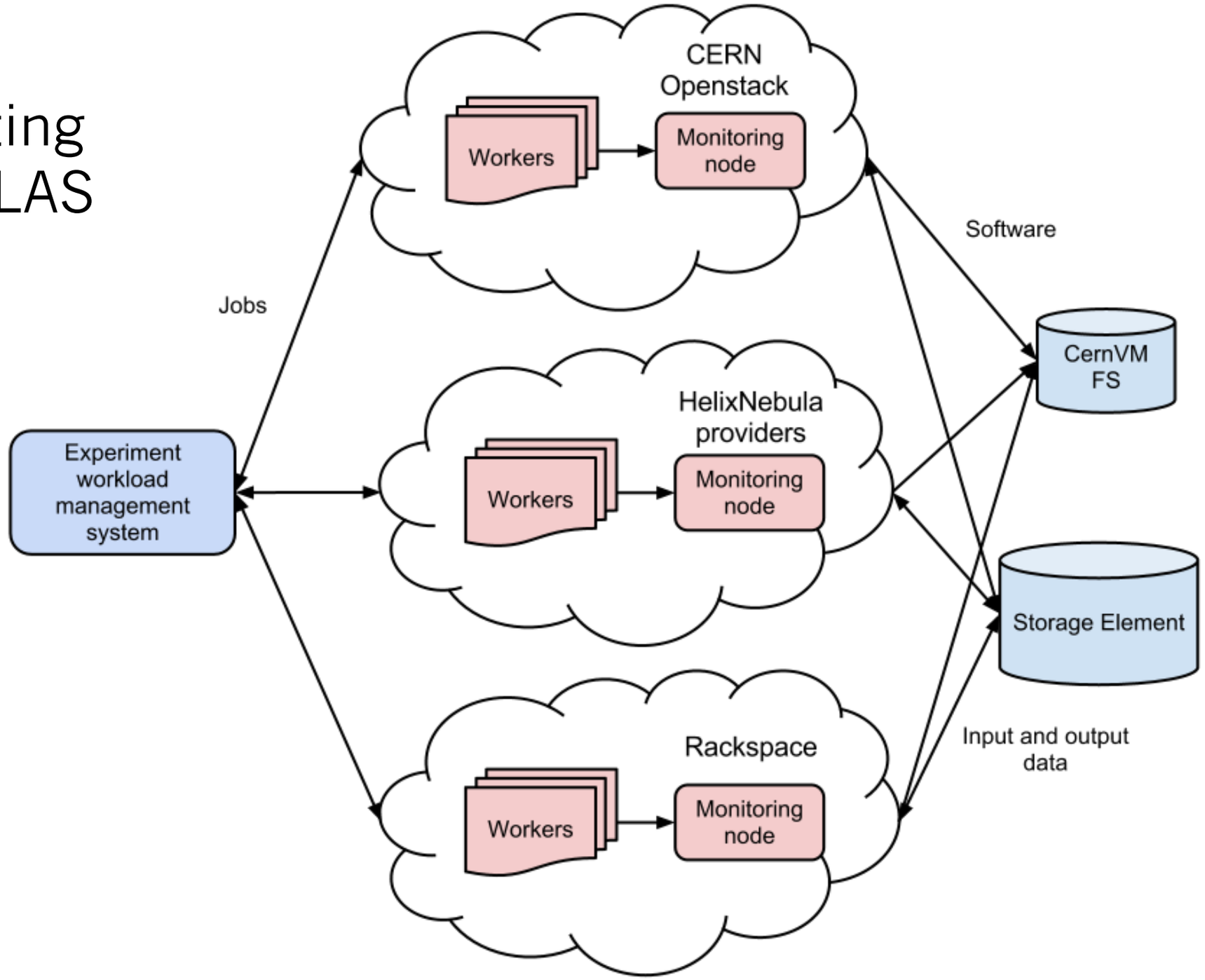
- **Marek** Denis
- **Cristovao Jose** Domingues Cordeiro
- **Katarzyna** Kucharczyk
- **Ramon** Medrano Llamas
- *Myself*

Outline

- AgileInfrastructure testing
- Contextualization
- Deployment Machine/Job Features
- DeltaCloud
- HelixNebula

Schema

- Parallel testing between ATLAS and CMS



AgileInfrastructure Environment

- OpenStack based infrastructure
  - Folsom release
  - Both Nova and EC2 interfaces provided
- CernVM/SLC* images
  - CVMFS, Ganglia, Glidein bootstrap
  - but no dynamic contextualization (see later)
- Submission through WMAgent, CRAB-2 (HammerCloud)
  - (remote)GlideinWMS as scheduler automatically provisioning VM's
- CMS's quota == 800 cores

AI - HammerCloud

- Testing with standard Hammer Cloud jobs
  - Input data from EOS
  - Low scale (short jobs, small tasks)
  - But continuously submitting jobs
  - Since beginning of may

- First results collected
  - reliability/integration of components
  - performances

AgileInfrastructure feedback

- Infrastructure (now Ibex release) has sensibly improved since previous release (Hamster)
- During last month we had twice the situation where it was non possible to boot any more VM's
  - solved as soon as reported (JIRA)
- Some minor issues now and then
  - VM's stuck in build (always reported on JIRA)
- **CMS does not see ATLAS's I/O inefficiencies**
  - No need of deeper investigation from CMS side

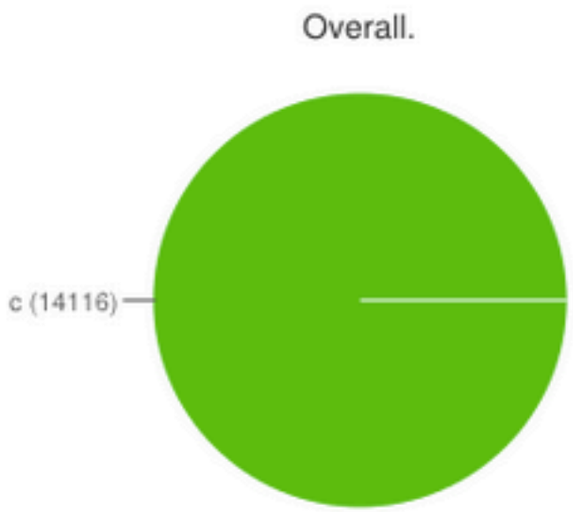
AI - GlideinWMS integration

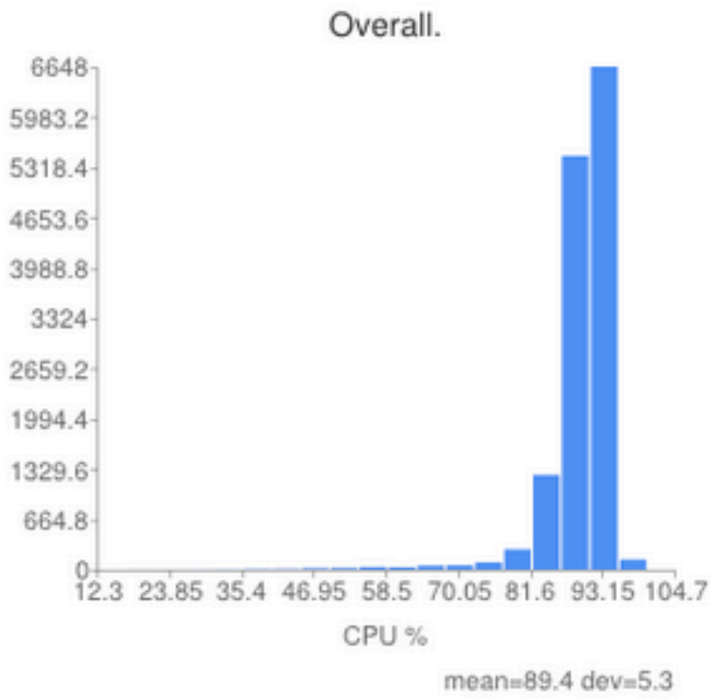
- Started with *Bing Bang approach*
  - integrated tools, discovered several internal issues in both sides; mostly solved;
  - GlideinWMS for cloud now more stable but still not production ready --- main issues:
    - too frequent polling
    - too many VM's being requested
    - loosing track of VM's
    - not correctly cleaning VM's
  - New release of HTCondor (9.7.6) should have fixed some of the issues
- We need login access to vocms172 to check logs

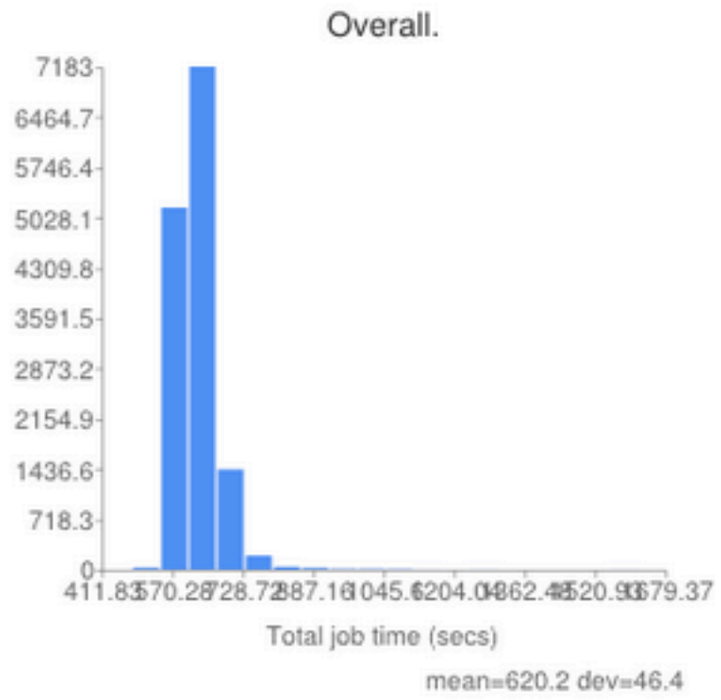
AgileInfrastructure performances

T2_CH_CERN_AI: job success rate 99.27%
T2_CH_CERN job success rate 86.13%

Input type: CMSSW_5_3_1
Input DS Patterns: /GenericTTbar/HC-CMSSW_5_3_1_START53_V5-v1/GEN-SIM-RECO
Ganga Job Template: glidein_test_ai.tpl
User code: pf2pat_cfg.py
Option file: empty
Template: glidein cloud AI
Logs

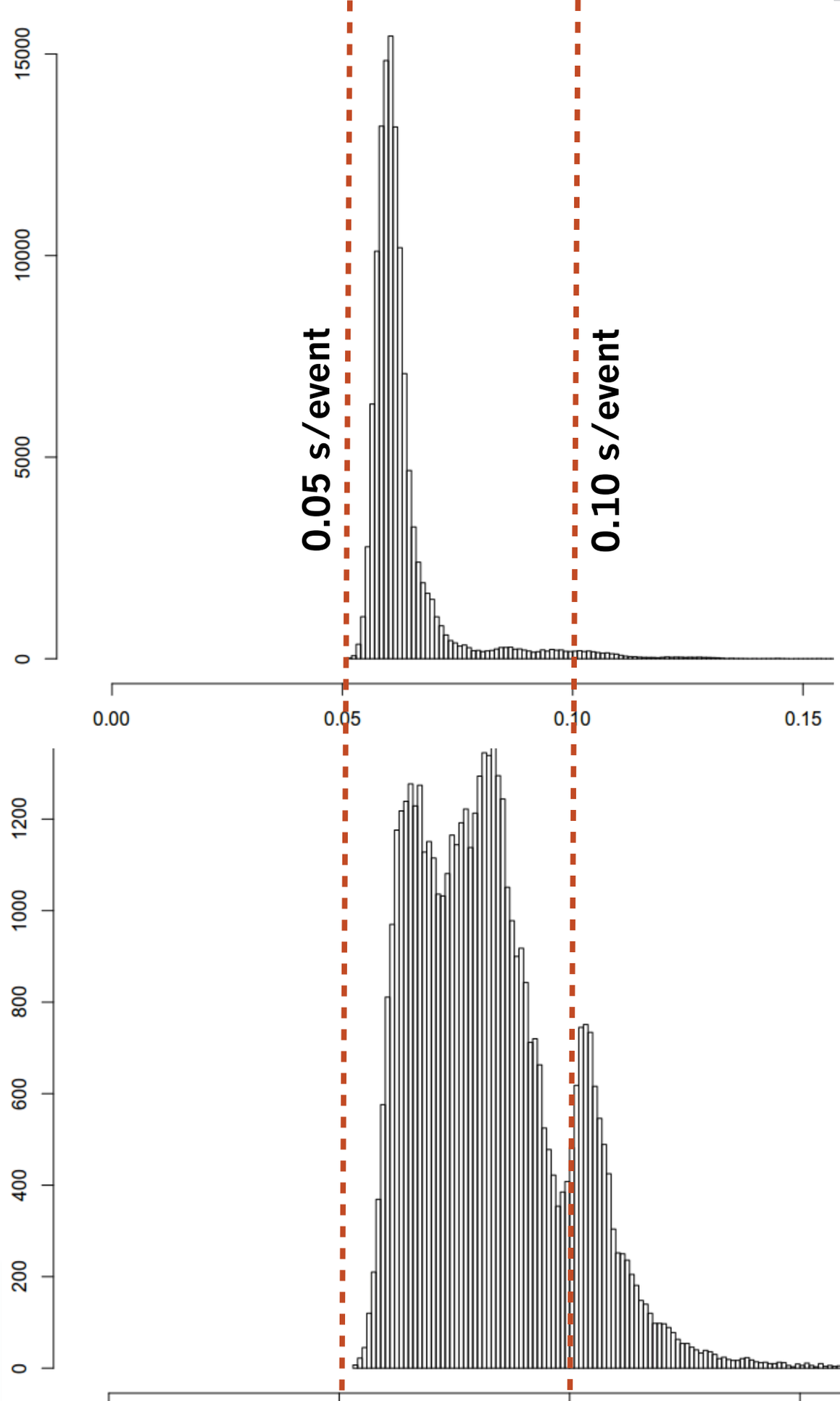






Time per event (s)

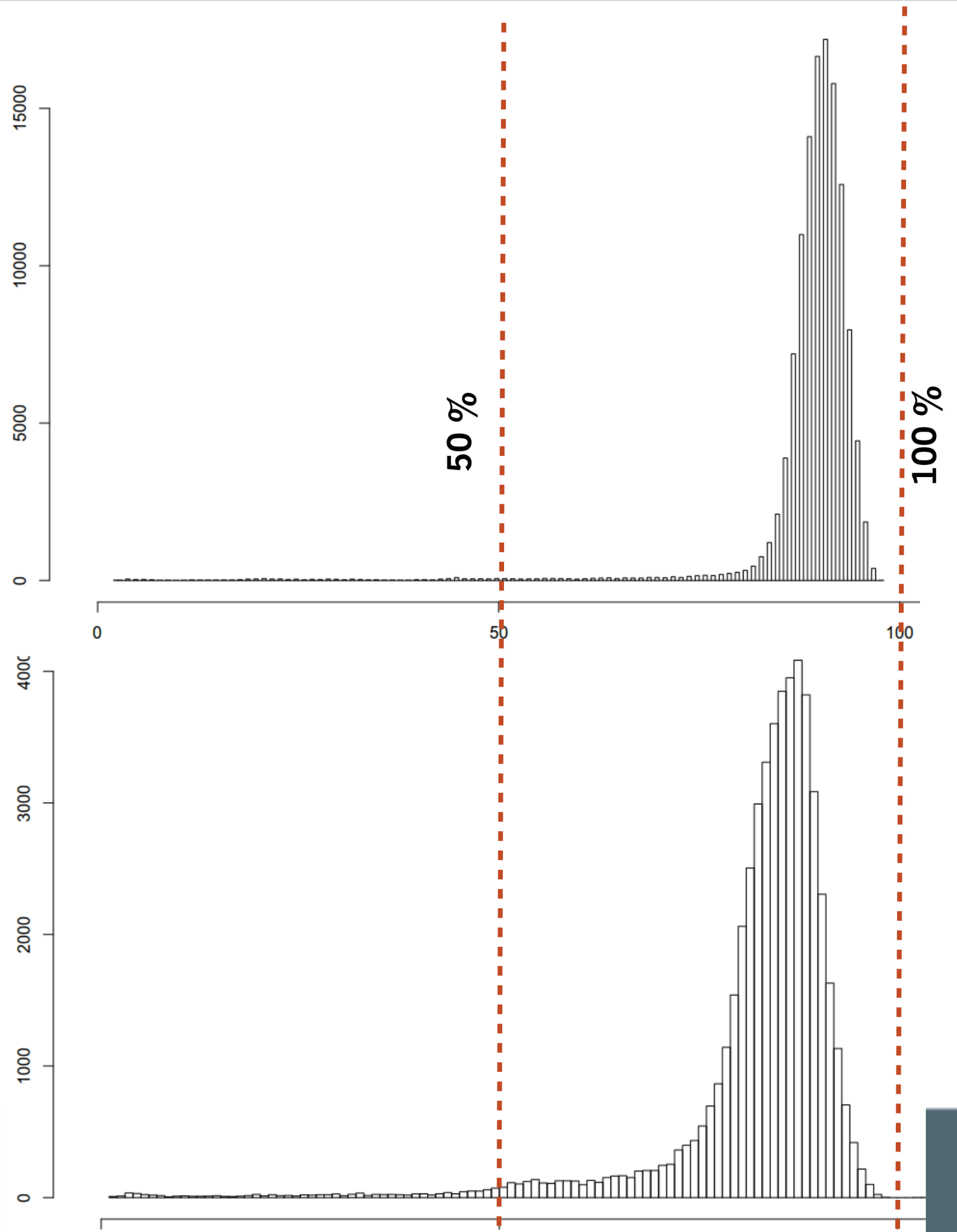
T2_CH_CERN_AI



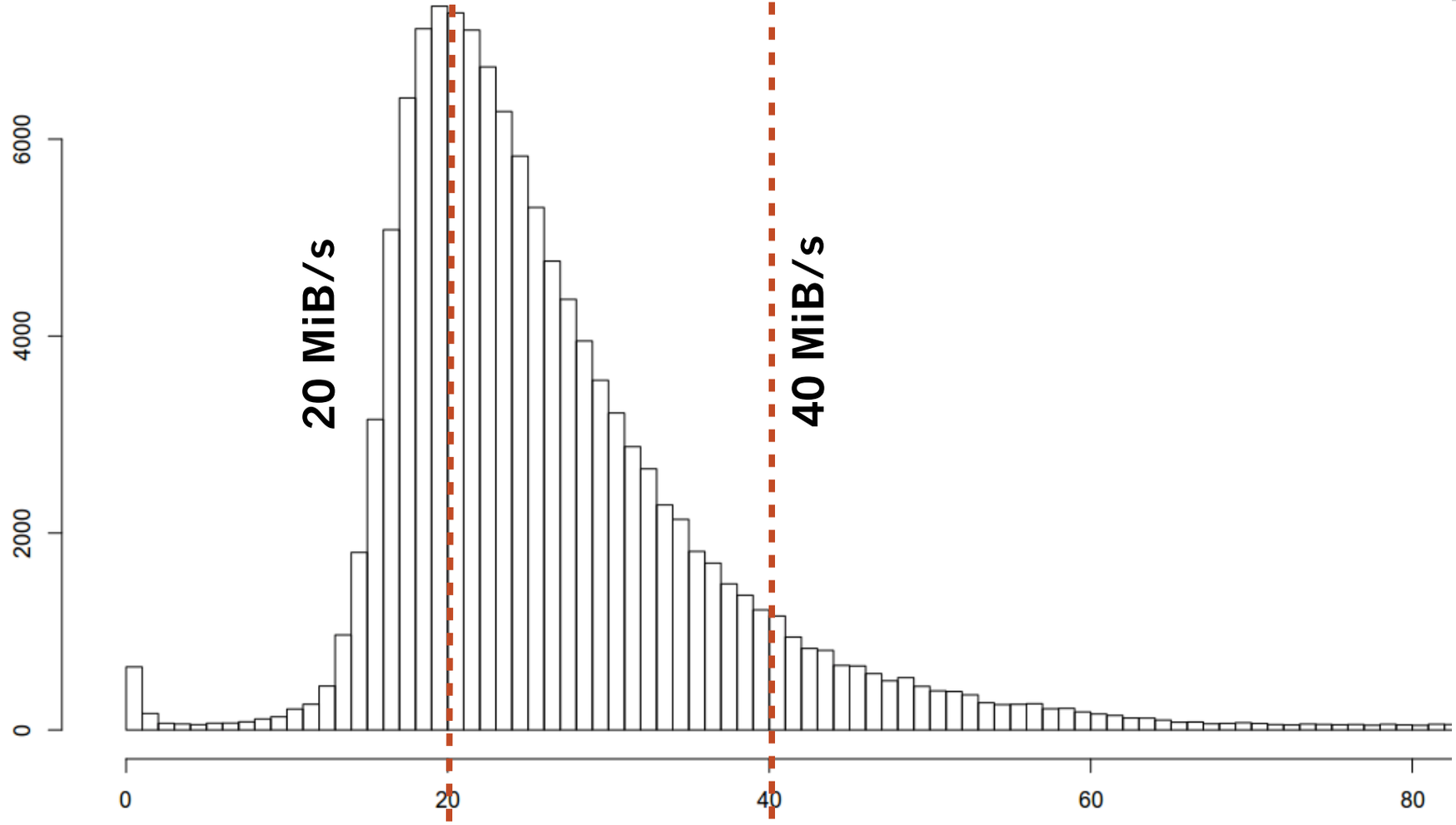
T2_CH_CERN

CPU %

T2_CH_CERN_AI

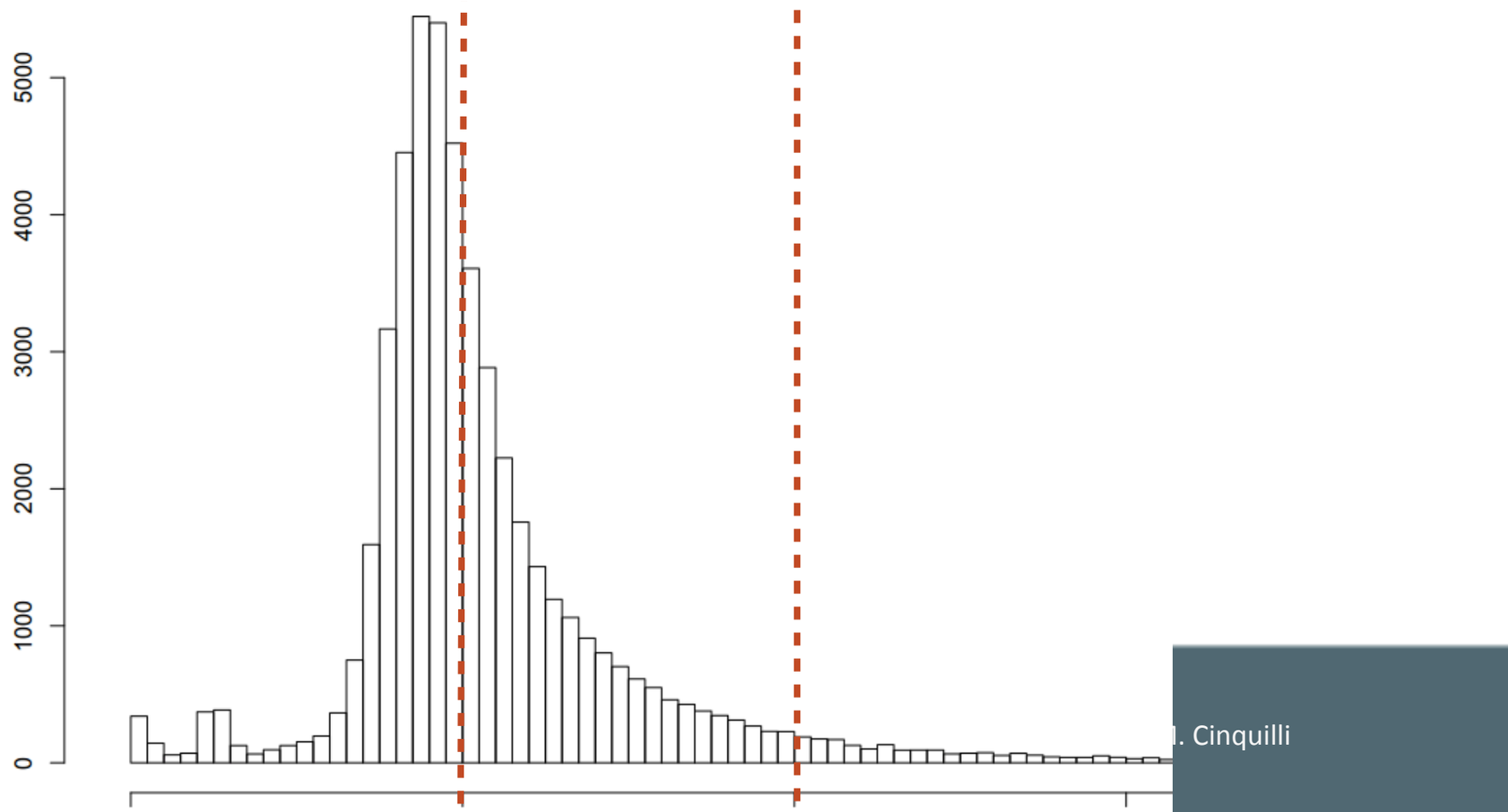


T2_CH_CERN



Average bandwidth (MiB/s)

T2_CH_CERN_AI



T2_CH_CERN

Contextualization

- Configuring an image according to dynamic variables
  - hostnames (squids, monitoring), software dependencies configurations, ...
- CernVM relies on AMI config scripts:
  - developed/maintained by CERN
- GlideinWMS has no dynamic contextualization
  - Golden image concept: everything is already configured, *glidein bootstrap* connects to frontend
  - If something needs to be changed, need a new image

CloudInit

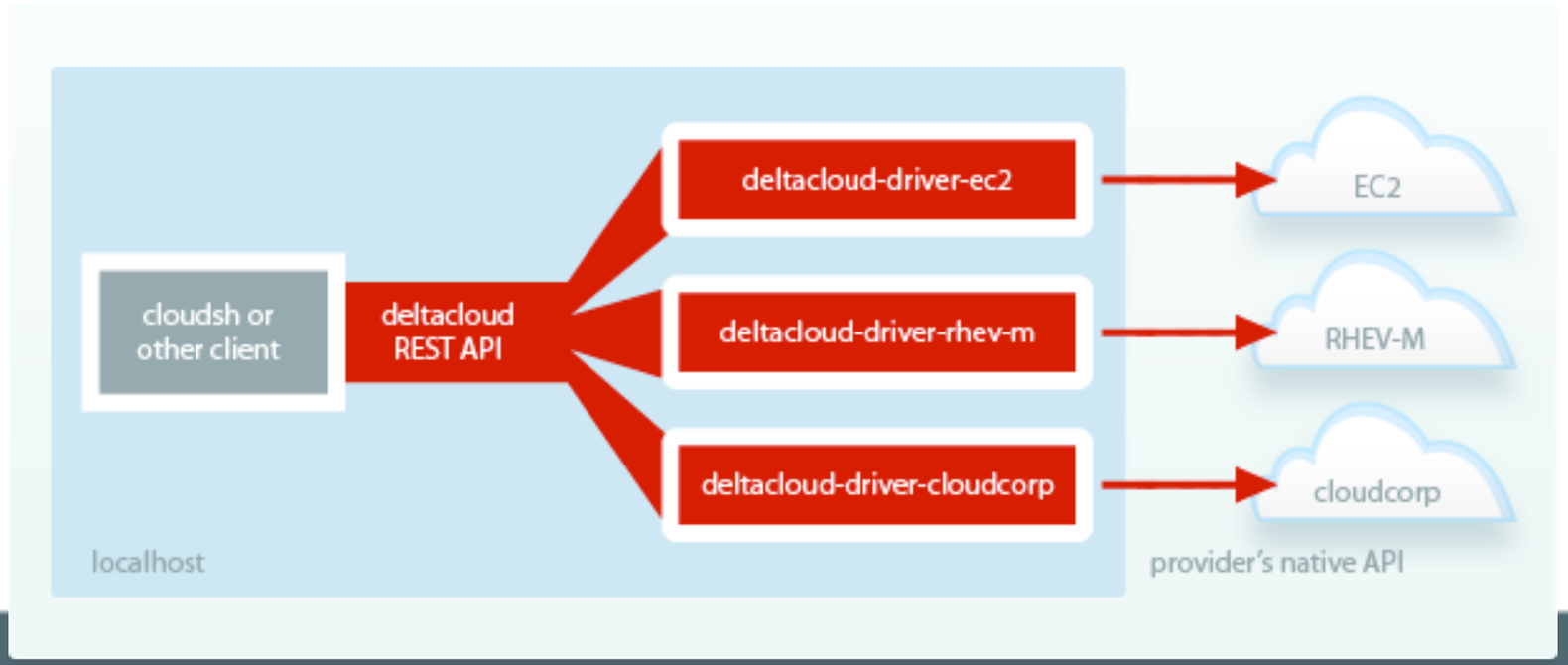
- Outside world is focusing on CloudInit to contextualize VirtualMachine
  - Ubuntu project, portable in ~every OS
  - 1 Simple plug-in for every module to be configured
- IT-SDC-OL developed the following modules:
  - condor, ganglia, cvmfs
  - *glidein bootstrap* (prototype)
- Documentation on how to use them
  - https://twiki.cern.ch/twiki/bin/view/LCG/CloudInit
- CernVM 3.X will possibly include these

Deployment Machine/Job Features

- Proposed the deployment of a communication layer between VO and resource provider
  - providing individual information about the worker node and job constraints to the VO pilot
  - More on Stefan Roiser GDB talk: http://indico.cern.ch/getFile.py/access?contribId=8&sessionId=0&resId=0&materialId=slides&confId=197804
- Cloud use cases:
  - Drain an hypervisor for dynamic rolling interventions on a site
  - Notify all machines in HLT farms to shutdown
  - Rebalancing of VO share

DeltaCloud

- Unique interface to interact with a wide range of cloud interfaces (Eucalyptus, OpenStack, Rackspace, ...)
- Exposes 3 different APIs: DeltaCloud, EC2, DMTF CIMI
- Status:
  - first installation done
  - functional evaluation looks good
  - able to interact with AI nova API



HelixNebula

- CMS joined ATLAS in the PoC2 since this year
- Evaluating two main solutions developed to federate several cloud providers
  - codename: BlueBox
- EnStratius (now owned by Dell):
  - first evaluation done
  - few slots (40 cores) available for a short period (2 weeks)
- Slipstream (developed by SixSq)
  - about to start the evaluation
  - waiting for images and accounts

Next Steps

- Increase the scale of AI testing through HammerCloud (GlideinWMS/EC2)
  - need longer jobs, but still comparable to other site results
- Remote data center (Wigner) will also need to be tested
- Waiting for stable GlideinWMS and newer policies on machine booting
  - Grid: *"conquer the resources"*
  - Cloud: *"ask for resources"*
- Looking forward to operation team using AI productively