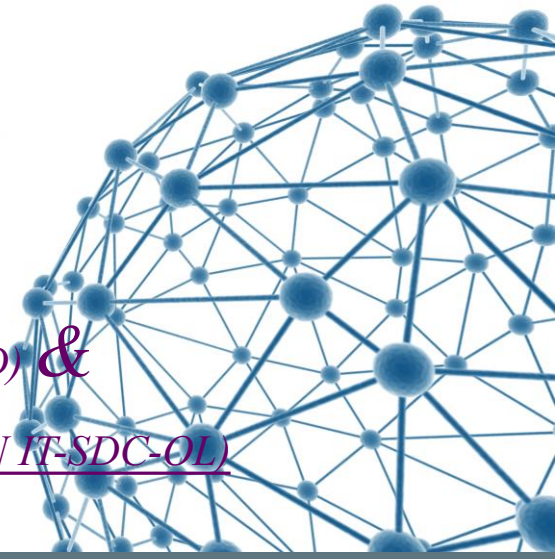



ATLAS & CMS Online Clouds

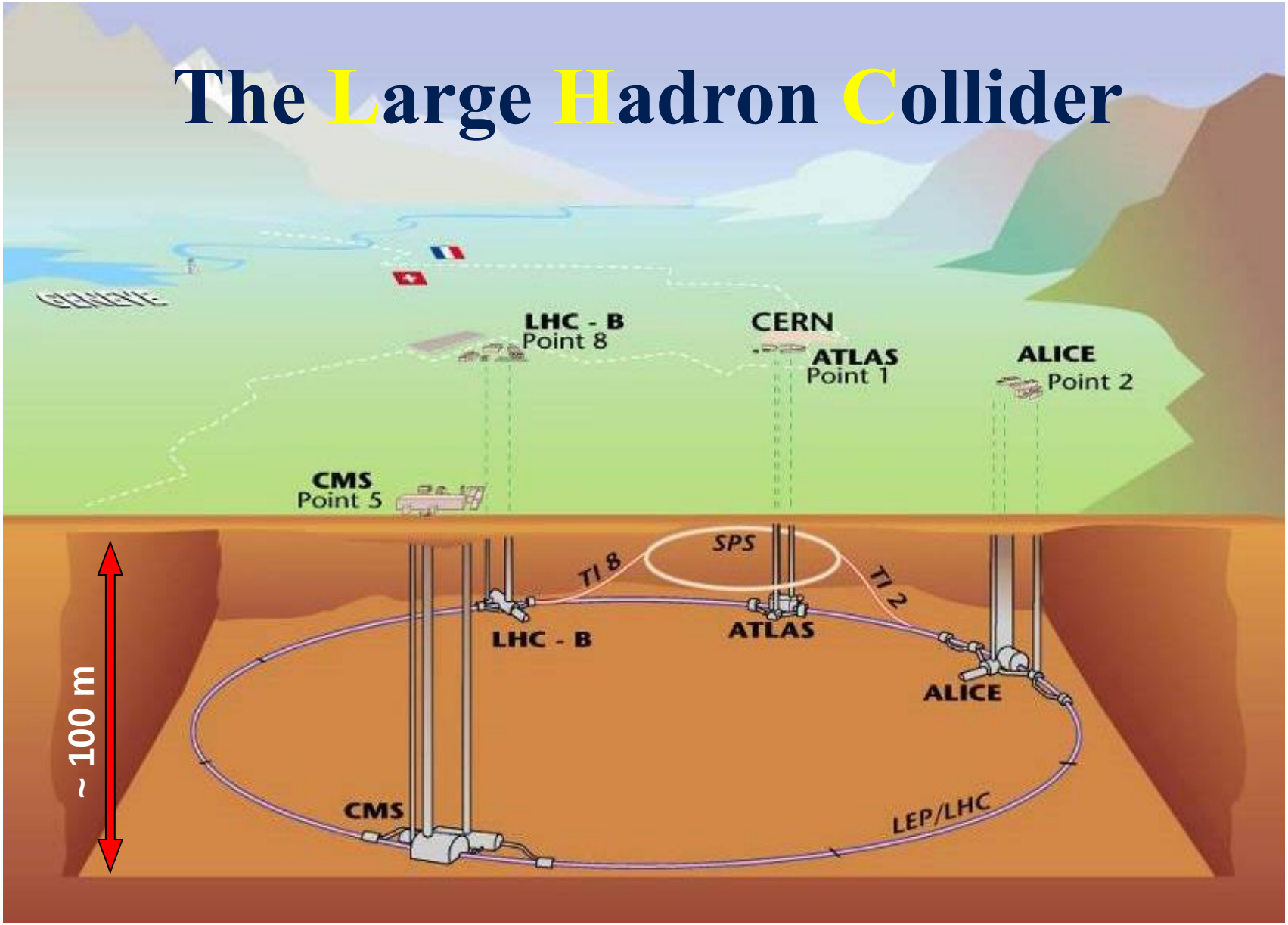
✓ exploit the Experiments' online farms for offline activities

during the LS1 & beyond

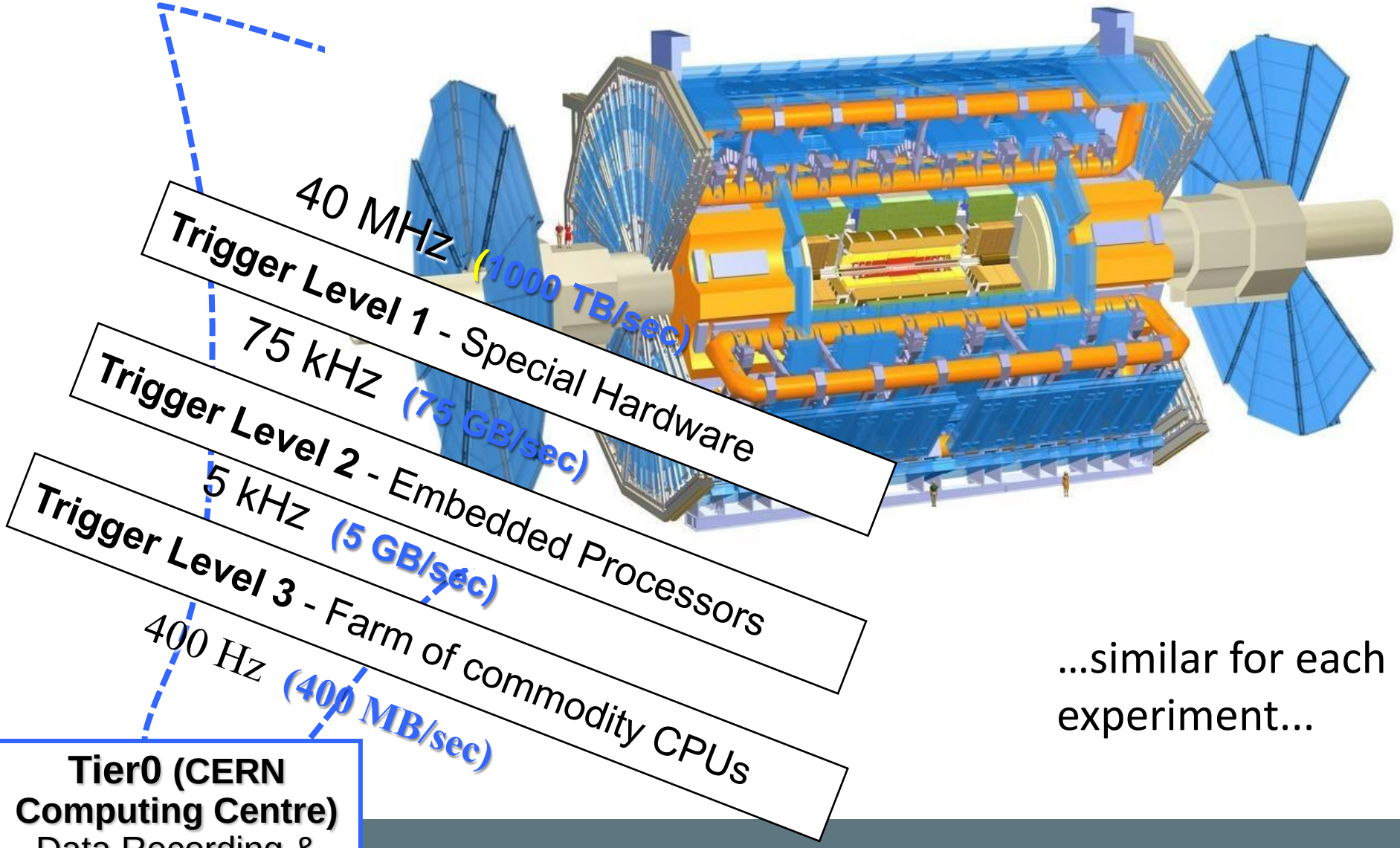


Olivier Chaze *(CERN-PH-CMD)* &
Alessandro Di Girolamo *(CERN IT-SDC-OL)*

The Large Hadron Collider



The experiment data flow



...similar for each experiment...

Resources overview

- High Level Trigger Experiment farms
  - ATLAS P1:
    - 15k cores (28k Hyper Threading, 25% reserved for TDAQ)
  - CMS P5:
    - 13k cores (21k Hyper Threading)
  - ! when available:
    - ! ~50% bigger than the Tier0,
    - ! doubling the capacity of biggest Tier1 of the Experiments

- Network connectivity to the IT Computing Centre (Tier0)

Type	Current status
P1 ↔ CERN IT CC (so called Castor link)	**70 Gbps (20 Gbps reserved for Sim@P1)**
P5 ↔ CERN IT CC	**20Gbps (80 Gbps foreseen in the next months)**

Why

- Experiments always resource hungry:
  - ATLAS + CMS: more than 250k jobs running in parallel

- … exploit all the available resources!

Teamwork

- Experts from the Trigger & Data Acquisition teams of the Experiments
- Experts from other institutes
  - BNL RACF, Imperial College …
- Experts of WLCG (Worldwide LHC Computing Grid)

Why Cloud?

- Cloud as an overlay infrastructure
  - provides necessary management of VM resources
  - support & control of physical hosts remain with TDAQ
  - delegate Grid support
- easy to quickly switch from HLT ↔ Grid
  - during LS1: periodic full-scale test of TDAQ sw upgrade
  - can be used in the future also during short LHC stop
- OpenStack: common solution, big community!
  - CMS, ATLAS, BNL, CERN IT….
  - sharing experiences
  - …and support if needed

OpenStack

- Glance: VM base image storage and management
  - Central image repository (and distribution) for Nova
- Nova: Central operations controller for hypervisors and VMs
  - CLI tools, VM scheduler, Compute node client
    - Network in multi-host mode for CMS
- Horizon/ High level control tools
  - WebUI for Openstack infrastructure/project/VM control (limited use)
- RabbitMQ

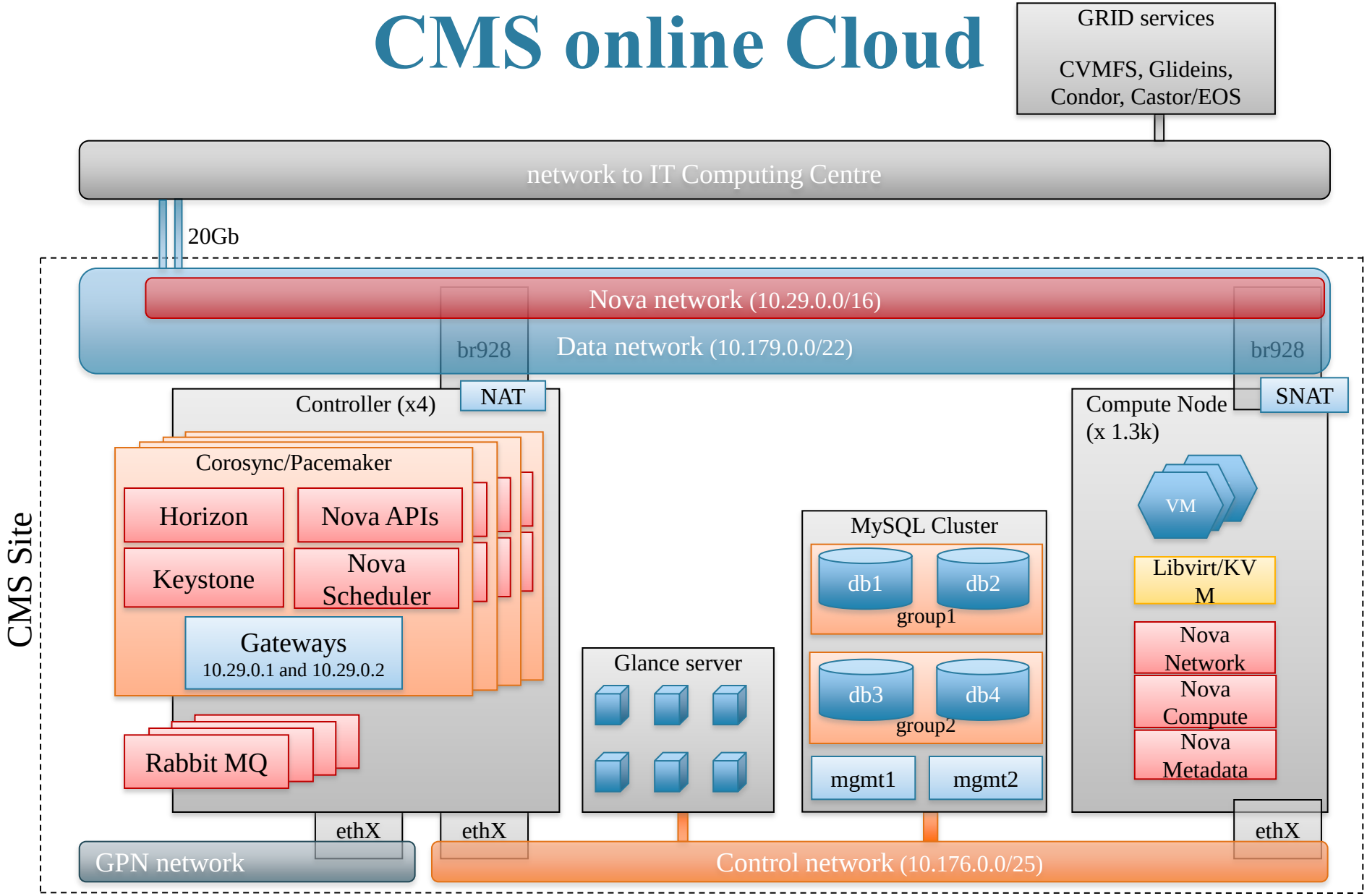
- ATLAS: OpenStack version currently used: **Folsom**
- CMS: OpenStack version currently used: **Grizzly**

Network challenges

- Avoid any interference with:
  - Detector Control System operations
  - internal Control network
- Each compute node is connected to two networks
  - One subnet per rack per network
  - Routers allow traffic to registered machines only

- ATLAS
  - A new dedicated VLAN has been setup
    - VMs are registered on this network
- CMS
  - VMs aren't registered
  - SNAT rules defined on the hypervisors to bypass network limitations.

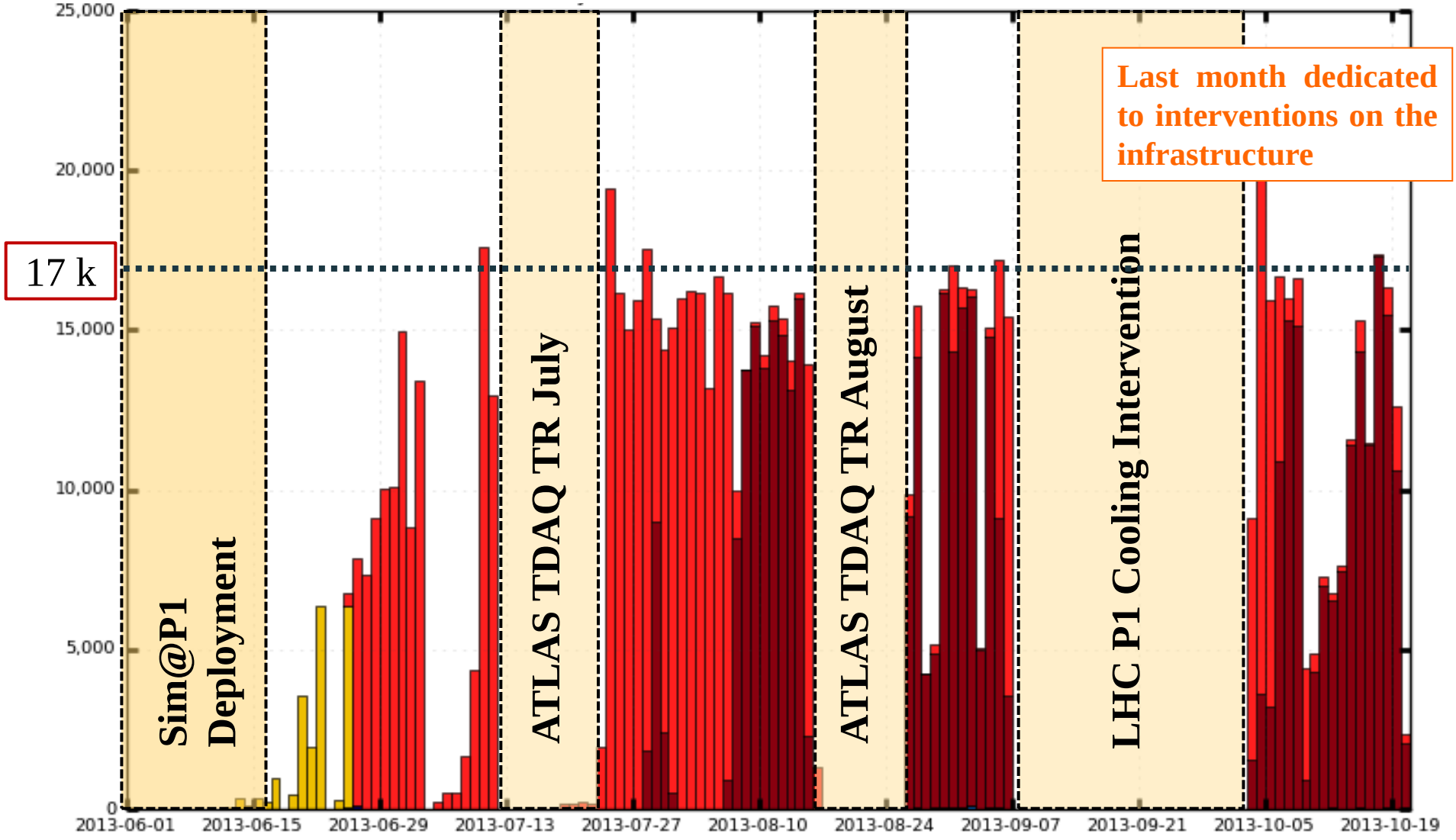
CMS online Cloud



VM image

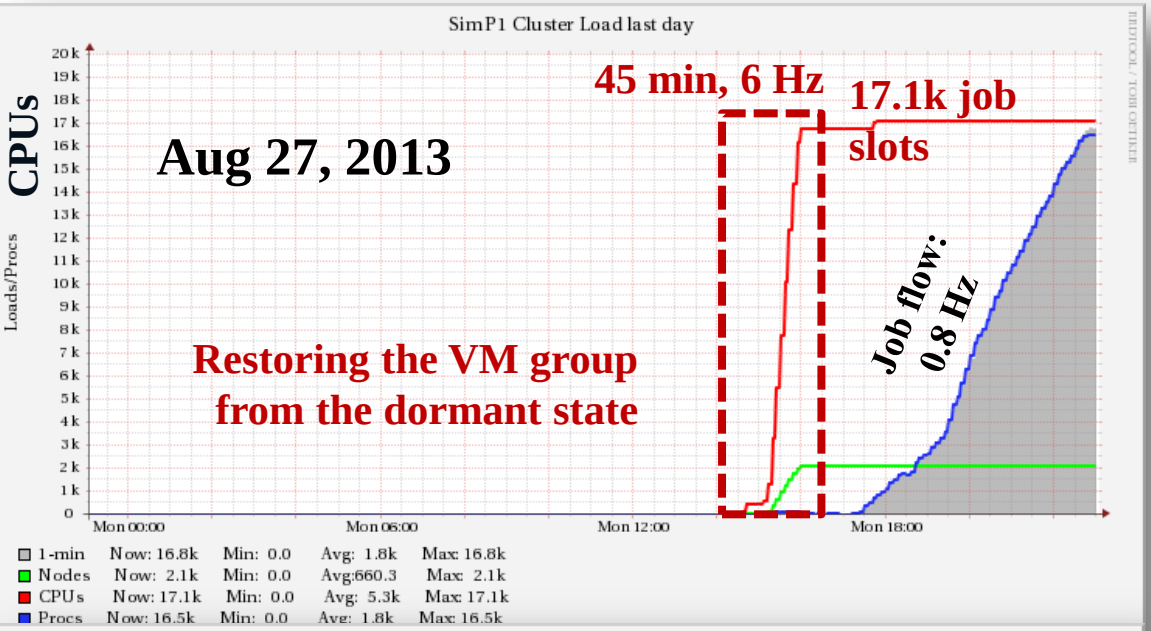
- SL5 x86_64 based (KVM HyperVisor)
  - Post-boot contextualization method: script injected into the base image (Puppet in the future)
  - Pre-caching of images on HyperVisors
    - bzip2 compressed QCOW2 images that are about 350 MB
- Experiment specific SW distribution with CVMFS
  - CVMFS: network file system based on HTTP and optimized to deliver experiment software.

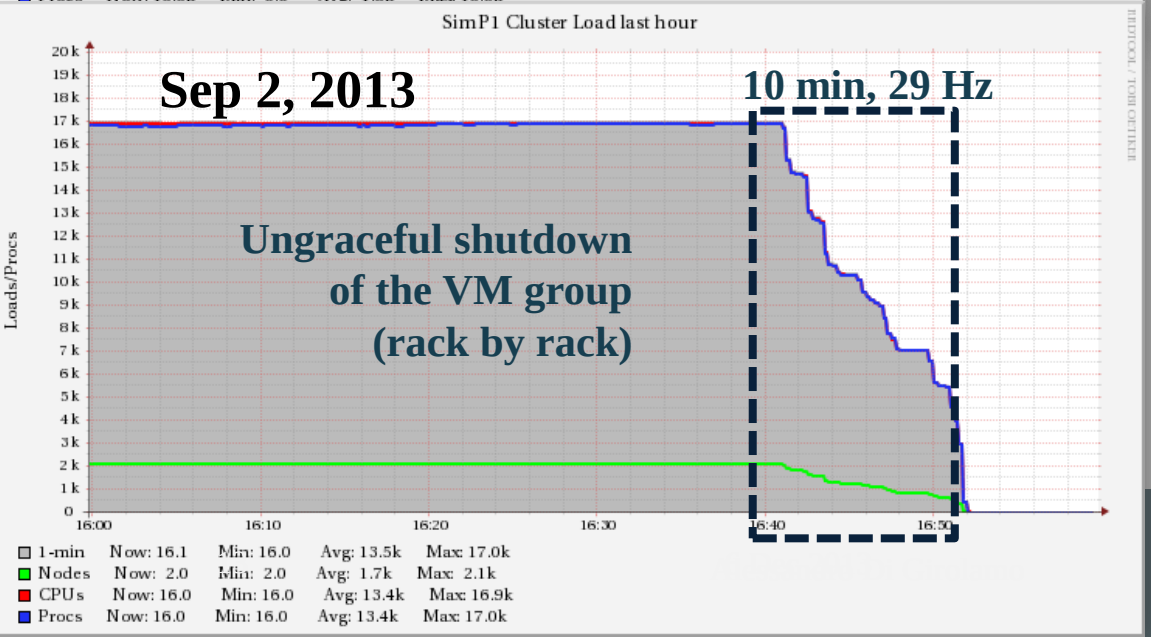
ATLAS: Sim@P1 running jobs: 1 June – 20 Oct



Overall: ~ 55% of time available for Sim@P1

ATLAS: Sim@P1: Start & Stop

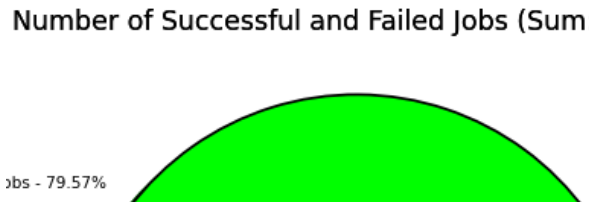




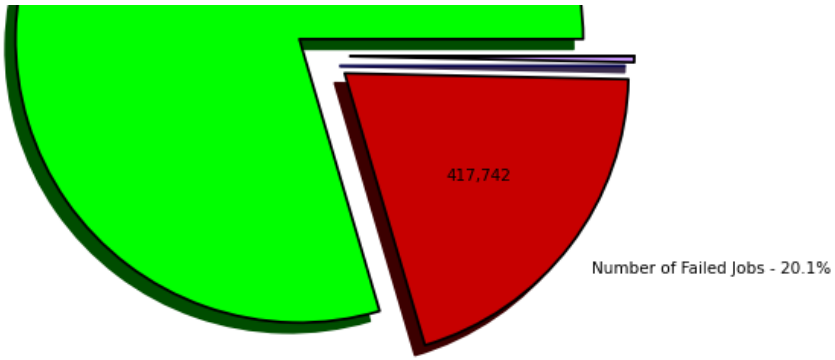
- Restoring Sim@P1:
  - VMs all up and running within 45min (6Hz)
  - MC Production jobs flow 0.8 Hz
    - ✓ now improved to almost 1.5 Hz
- Shutdown:
  - 10min (29Hz) the infrastructure is back to TDAQ

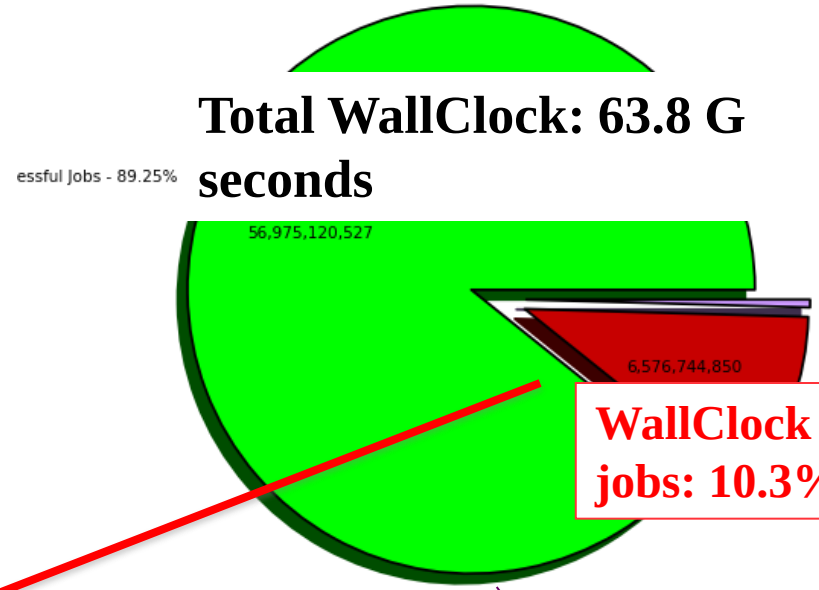
ATLAS: Sim@P1 completed jobs: 1 June – 20 Oct

Overall: ~ 55% of time available for Sim@P1



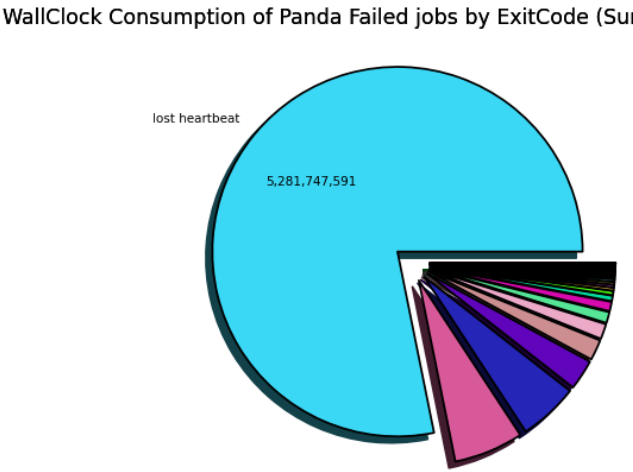
Total successful jobs: 1.65M
Efficiency: 80%





Total WallClock: 63.8 G seconds

WallClock failed jobs: 10.3%



78% Lost Heartbeat:
Intrinsic to the opportunistic nature of resources

Comparison with **CERN-PROD**
Total WallClock: 83.3 Gsec
WallClock failed jobs: 6%

Conclusions

- ✓ Experiments online Clouds are a reality
  - ✓ Cloud solution: no impact on data taking, easy switch of activity:
    - ✓ Quick onto them, quick out from them, e.g.:
      - ! from 0 to 17k Sim@P1 jobs running in 3.5hours
      - ! from 17k Sim@P1 jobs running to TDAQ ready in10 mins
  - ! contributing to computing as one big Tier1 or CERN-PROD!

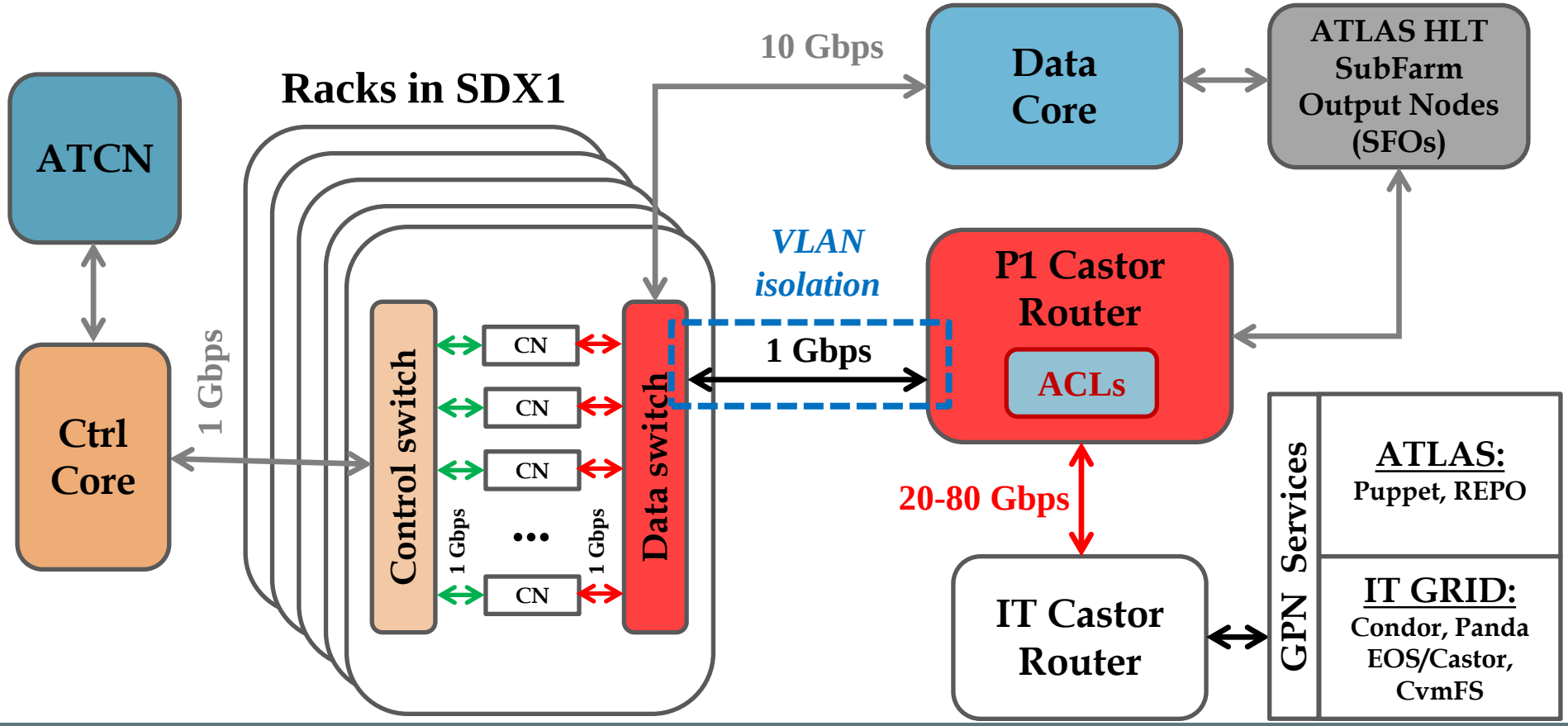
- ✓ Operations: still a lot of (small) things to do
  - ! Integrate OpenStack with the online control to allow dynamic allocation of resources to the Cloud
  - ! open questions not unique to the experiments' online clouds: opportunity to unify solutions to minimize manpower!

BackUp

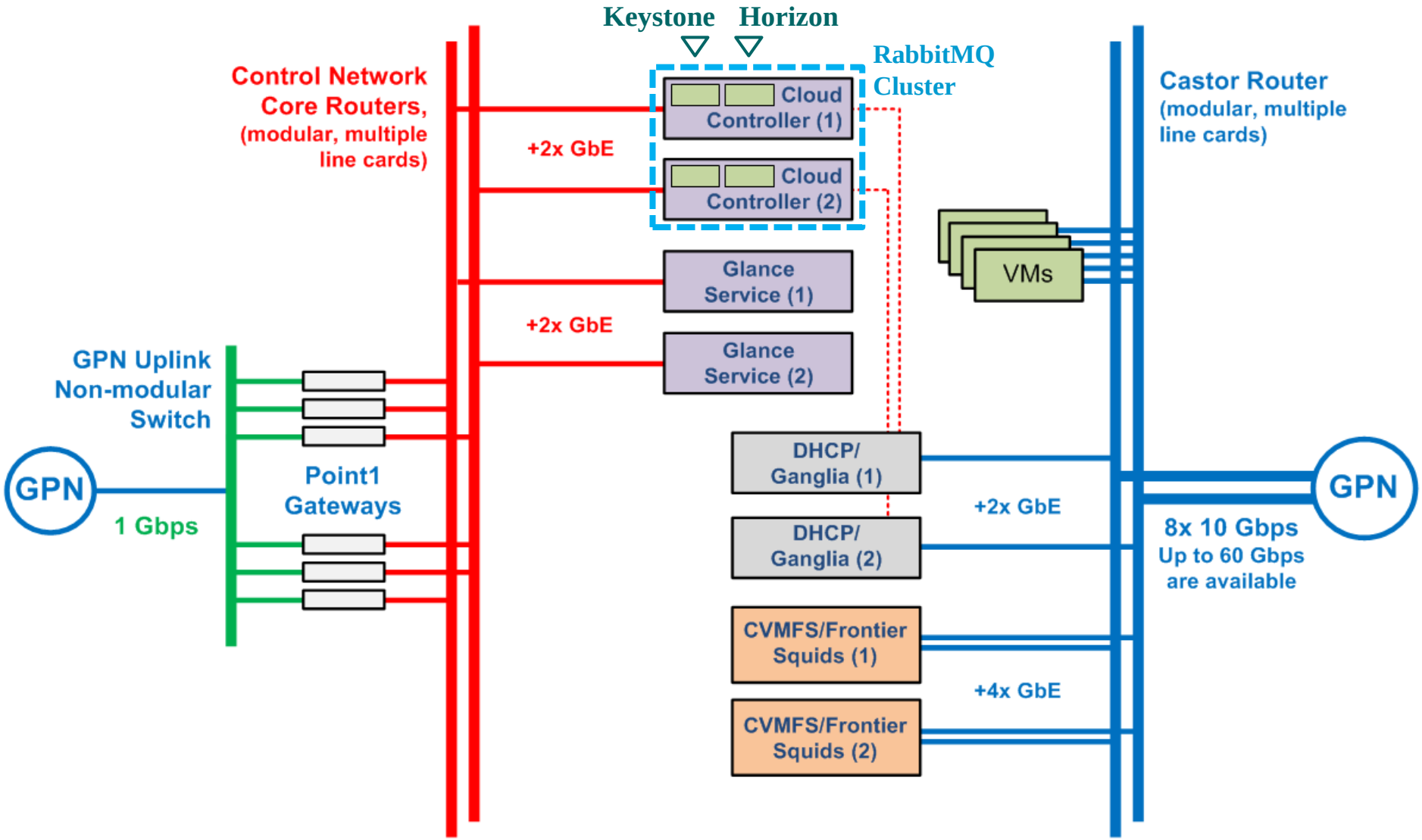

MY REPORT COMES TO THE CONCLUSION THAT CLOUD TECHNOLOGY IS OF NO USE TO THIS COMPANY. I'LL UPLOAD IT TO DROP BOX SO YOU CAN TAKE A LOOK AT IT.
© D.Fletcher for CloudTweaks.com

Sim@P1: Dedicated Network Infrastructure

- ✓ Sim@P1 VMs will use a dedicated 1 Gbps physical network connecting the P1 rack data switches to the "Castor router"



Cloud Infrastructure of Point1 (SDX1)



Keystone (2012.2.4)

Slide from Alex Zaytsev (BNL RACF)

- No issues with stability/performance observed of any scale
- Initial configuration of tenant / users / services / endpoints might deserve some higher level automation
  - Some automatic configuration scripts were already available in 2013Q1 from the third parties, but we found that using the Keystone CLI directly is more convenient & transparent
- Simple replication of the keystone MysQL DB works fine for maintaining redundant Keystone instances

Nova (2012.2.4)

Slide from Alex Zaytsev (BNL RACF)

- Once bug fix needed to be applied
  - Back port from Grizzly release:
    "Handle compute node records with no timestamp":
    https://github.com/openstack/nova/commit/fad69df25ffcea2a44cbf3ef636a68863a2d64d9
- The prefix for the VMs' MAC addresses had to be changed in order to match the range pre-allocated for Sim@P1 project
  - No configuration option for this, direct patch to Python code was needed
- Configuring the server environment for Nova Controller supporting more than 1k hypervisors / 1k of VMs requires rising the default limits for maximum number of open files per several system users
  - Not documented / handled automatically by Openstack recommended configuration procedures, but pretty straightforward to figure out
- RabbitMQ cluster consisting of minimum two nodes was required in order to scale beyond 1k hypervisors per single Nova Controller
  - RabbitMQ configuration procedure / stability is version sensitive
  - We had to try several version (currently v3.1.3-1) before achieving a stable cluster configuration
- Overall: stable long term operations with only one Cloud controller (plus one hot spare backup instance) for the entire Point 1

Glance (2012.2.4)

Slide from Alex Zaytsev (BNL RACF)

- Single Glance instance (provided with single 1 Gbps uplink) works nicely as an central distribution point up to the scale of about 100 hypervisors / 100 VM instances
- Scaling beyond that (1.3k hypervisors, 2.1k VM instances) requires either
  - A dedicated group of cache servers between Glance and hypervisors
  - Custom made mechanism for pre-deployment of the base images on all compute nodes (multi-level replication)
- Since we operate with only one base image at the time which changes rarely (approximately once a month) we built a custom image deployment mechanism, living the central Glance instances with functionality of image repositories, but not the image central distribution points
  - No additional cache servers needed
  - We distribute bzip2 compressed QCOW2 images that re only about 350 MB in size
  - Pre-placement of the new image to all the hypervisors take in total only about 15 minutes despite 1 Gbps network limitations on both Glance instances and on the level of every rack of compute nodes
- Snapshot functionality of Glance is used only for making persistent changes in the base image
  - No changes are saved for VM instances during production operations

Horizon (2012.2.4)

Slide from Alex Zaytsev (BNL RACF)

- Very early version of the web interface
- Many security features are missing
  - Such as no native HTTPS support
- Not currently used for production at Sim@P1
- Several configuration / stability issues encountered
  - Such as debug mode must be enabled in order for Horizon to function properly
- Limited feature set of the web interface
  - No way to perform non-trivial network configuration purely via web-interface
  - No way to handle large groups of VMs (1-2k+) in a conveniently, such as to display VM instances in a tree structured according to the configuration of the availability zones / instance names
  - No convenient way to perform bulk operations on large subgroups of VMs (hundreds) within the production group of VMs consisting of 1-2k
- All of these problems, presumably, already addressed in the resent Openstack releases

. . .

- . . .
  - . . .
  - . . .
- . . .
  - . .
  - . . .
  - . .
    - ! . . .

. . .

- . . .
  - . . .
  - . . .
- . . .
  - . .
  - . . .
  - . .
    - ! . . .