Opportunistically turning the HLT farm into a cloud

Dario Berzano

CERN

ALICE Offline Week - CERN, 26.06.2014

Forthcoming HLT production farm

- This fall: ~180 nodes with 2 Intel Xeon processors, 8 or 10 cores each
- Current test nodes are ASUS ESC4000 FDR G2:
  - 2 Intel Xeon E5-2690 with 10 cores each at 3 GHz
    - 20 cores per node (40 threads with hyperthreading)
  - 128 GB RAM
    - 6,4 GB per core (3,2 GB with hyperthreading)
  - GPU: AMD Firepro W8000 graphics card
  - SSD disks mirrored in RAID
  - 1 GbE + InfiniBand
- Uplink to the CERN General Purpose Network: 80 Gbit/s

Rationale: do not waste resources

- HLT farm is very powerful
  - ~5000÷7000 job slots (with hyperthreading)
  - Can be compared to a Tier-1
- HLT resources not used all the time
  - shutdowns, technical stops, between fills
  - or during a run part of HLT might be unused
- Use HLT resources for executing Grid(-like) jobs
  - Already successfully pursued by ATLAS and CMS
  - ATLAS: WCT efficiency comparable to the Grid *(→ CHEP 2013)*

A private cloud on the HLT farm

- HLT farm is a delicate real-time environment
    - Opportunistic exploitation can by no means interfere with standard HLT operations
- Hard separation of HLT environment and the opportunistic one
    - Best isolation technique: configure HLT nodes as a private cloud
- We start working on the current "devel" farm
    - Configuration will be moved to the forthcoming "production" farm
    - We are considering OpenStack → popular, lots of support
- Ideal type of opportunistic jobs: CPU-intensive → Monte Carlos
    - I/O uplink and gateway might be a bottleneck on HLT

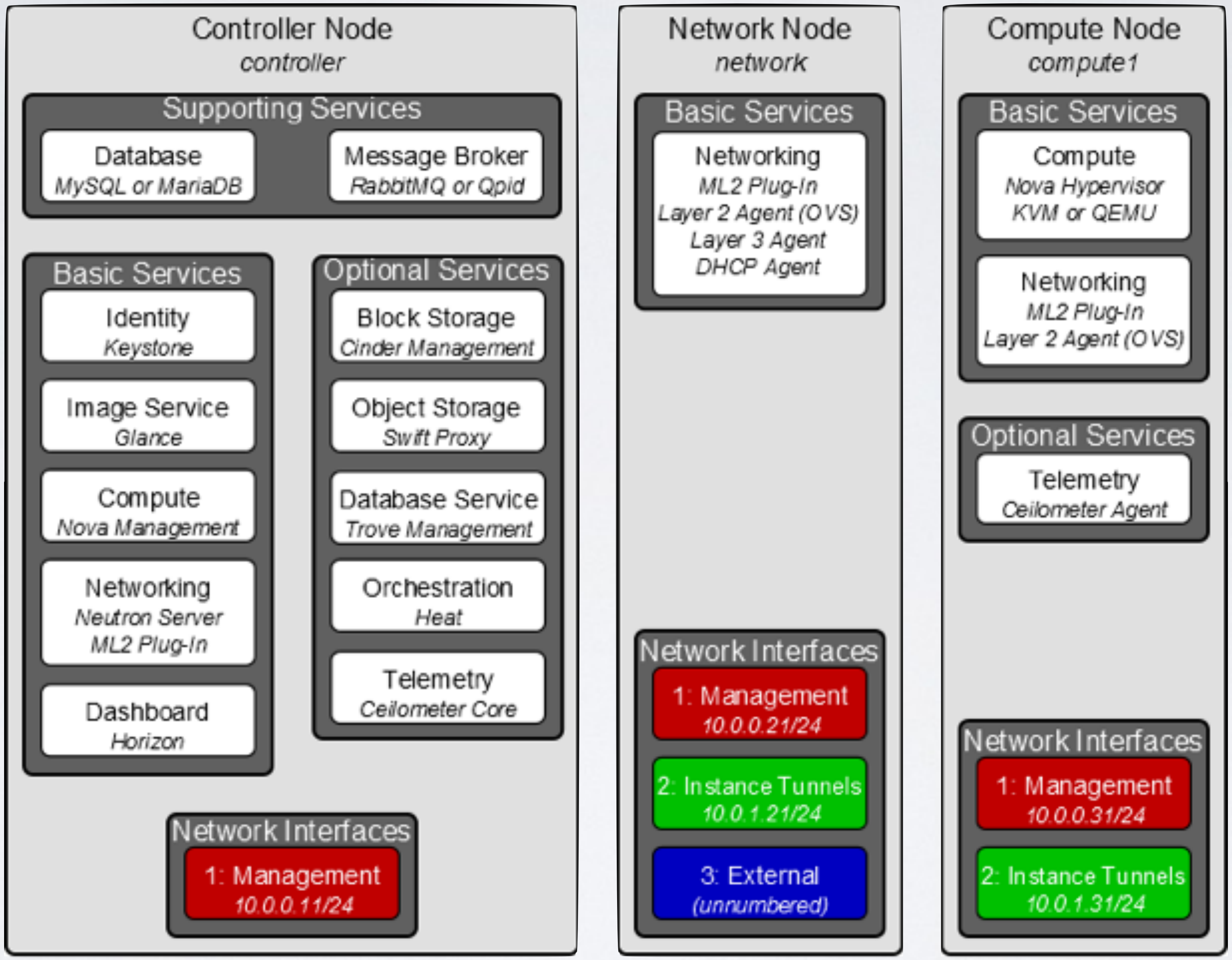
Proposed OpenStack configuration

Dedicated non-HLT node

Gateway

HLT nodes

Will also run an AliEn VOBOX



Controller Node
controller

Supporting Services

- Database — *MySQL or MariaDB*
- Message Broker — *RabbitMQ or Qpid*

Basic Services

- Identity — *Keystone*
- Image Service — *Glance*
- Compute — *Nova Management*
- Networking — *Neutron Server*, *ML2 Plug-In*
- Dashboard — *Horizon*

Optional Services

- Block Storage — *Cinder Management*
- Object Storage — *Swift Proxy*
- Database Service — *Trove Management*
- Orchestration — *Heat*
- Telemetry — *Ceilometer Core*

Network Interfaces

- 1: Management — *10.0.0.11/24*

Network Node
network

Basic Services

- Networking — *ML2 Plug-In*, *Layer 2 Agent (OVS)*, *Layer 3 Agent*, *DHCP Agent*

Network Interfaces

- 1: Management — *10.0.0.21/24*
- 2: Instance Tunnels — *10.0.1.21/24*
- 3: External — *(unnumbered)*

Compute Node
compute1

Basic Services

- Compute — *Nova Hypervisor*, *KVM or QEMU*
- Networking — *ML2 Plug-In*, *Layer 2 Agent (OVS)*

Optional Services

- Telemetry — *Ceilometer Agent*

Network Interfaces

- 1: Management — *10.0.0.31/24*
- 2: Instance Tunnels — *10.0.1.31/24*

Three-node architecture: bit.ly/os3nodes

Layers of isolation

- Running environment
  - All opportunistic jobs run inside virtual machines
  - The KVM hypervisor provides isolation
- Network
  - HLT has a private network
  - Virtual machines will have their own isolated network
    - Software-Defined Network with Open vSwitch and VLAN tagging
    - Hardware switches: traffic shaping → real-time priority to HLT

Administrative domains

- HLT operators:
  - Ultimate control on which HLT nodes are available as hypervisors
  - Via OpenStack:
    - suspend, resume, kill VMs - attach, detach hypervisors
  - In case of misbehavior of OpenStack, fallback to the "kill switch":
    - terminate target hypervisor's "compute" service
    - terminate VMs running on target hypervisors
- The Offline:
  - Run a special AliEn site on the virtual machines
  - Decides which jobs should be executed there

Configuration

- Full integration with current HLT management tools
  - Puppet and Foreman
- We start right away with Puppet
  - Abstract configuration details
  - First setup test: on the devel cluster
  - Easy to port them to the production cluster

Kill or suspend VMs?

- HLT notifies the Offline that resources will be reclaimed “soon”
  - Offline takes as many measures as possible to relinquish resources
  - In practice: with very long jobs (~10 h) this does not work
- Kill VMs, no matter if they are running jobs
  - AliEn: jobs will go to “zombie” and automatically resubmitted
  - If we only accept “short” jobs (1÷3 h) it might be acceptable
  - Little waste of resources and no special development
- Suspend VMs (and resume them later)
  - What runs inside the VMs might die anyway (proxies? I/O?)
  - Need development: AliEn must recognize a new “suspended” state

Milestones (preliminar)

- August:
  - base OpenStack services configured on the devel cluster
  - network isolation operational
- September:
  - test the devel configuration on the production cluster
  - network hardware configured for traffic shaping
  - configure the special AliEn VOBOX
- October:
  - ready for running special AliEn jobs

Thank you!