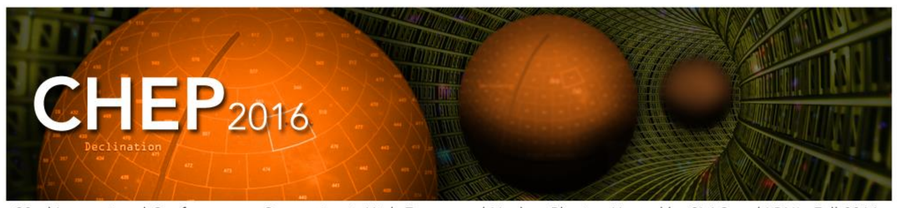
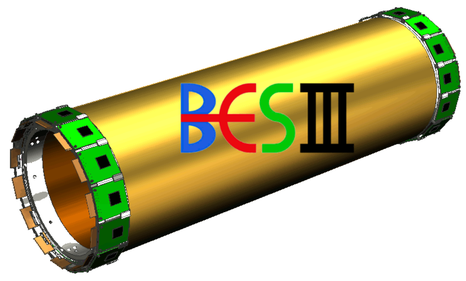


22nd International Conference on Computing in High Energy and Nuclear Physics, Hosted by SLAC and LBNL, Fall 2016



A modular (almost) automatic set-up for elastic multi-tenants cloud (micro)infrastructures

F. Astorino[a], A. Amoroso[a,b], S. Bagnasco[b], N. A. Balashov[c], F. Bianchi[a,b], M. Destefanis[a,b], M. Maggiora[a,b], J. Pellegrino[a,b], L. Yan[b], T. Yan[d], X. Zhang[d], X. Zhao[d]

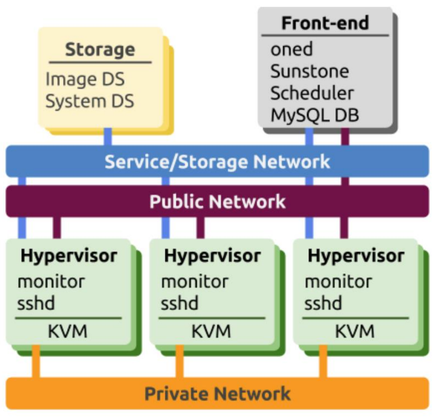
[a]University of Turin, [b]INFN-Turin, [c]Joint Institute for Nuclear Research (Dubna), [d]Institute of High Energy Physics, Chinese Academy of Sciences (Beijing)

OpenNebula

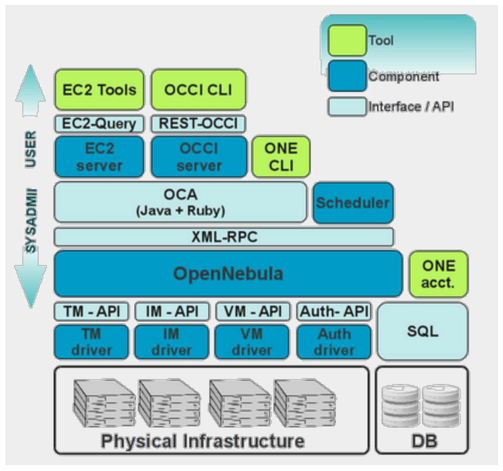
Physical hosts: servers hosting the Virtual Machines
- often called "Hypervisors" (like the software)
- KVM (OpenNebula supports also vCenter and Xen)
- monitoring daemons
- sshd for system connection

Networks
- Service and Storage network:
  - monitoring and control information
  - image transfers

Networks used by the Virtual Machines
- Private Network:
  - private IPs
  - intra-cloud communications
- Public Network:
  - public IPs
  - incoming connectivity to VMs



Storage
- Service datastores don't necessarily need to be shared across VMs:
  - images can be transferred to the hypervisors' disk through ssh and started locally
- Image Repository Datastore:
  - holds the OS images
- System Datastore
  - holds the running instances
  - if it's a shared FS, VMs can be "live-migrated"



Control node: runs the OpenNebula stack
- oned (the main daemon)
- schedd (the VM scheduler)
- Sunstone (the web-based GUI)
- MySQL DB backend (can be separate)
- API services (OCCI or EC2)
- advanced services (OneFlow, OneGate,…)
- control node unavailability does not affect running VMs
- only control on them (start & stop, monitoring,…)

VMDIRAC

"VM director" (vs DIRAC "Pilot director")
- starts VMs (vs DIRAC pilot jobs)

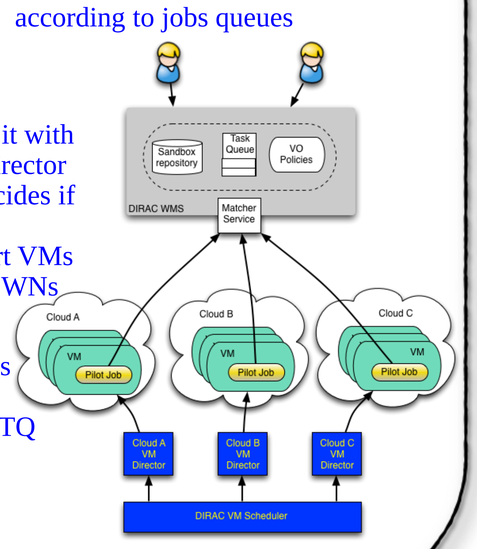
VMs at boot time start "pilot job"
- instantiated VMs behave just as other WNs w.r.t. DIRAC WMS

Integrate Federated cloud into DIRAC
- OCCI compliant clouds: OpenStack, OpenNebula
- CloudStack
- Amazon EC2

VMDIRAC architecture and components
- DIRAC server side
  - VM Scheduler – gets job status from TQ and match it with the proper cloud site, submits requests of VMs to Director
  - VM Manager – takes statistics of VM status and decides if needing new VMs
  - VM Director – connects with cloud manager to start VMs
  - Image context manager – contextualizes VMs to be WNs
- VM side
  - VM monitor Agent – periodically monitors the status of the VM and shutdowns VMs when no need
  - Job Agent – just like "pilot jobs", pulling jobs from TQ
- Configuration and load management
  - is used to configure the joined cloud and the image
  - starts VMS
  - runs jobs on VMs

VM scheduler
- manages dynamic virtual machines according to job status

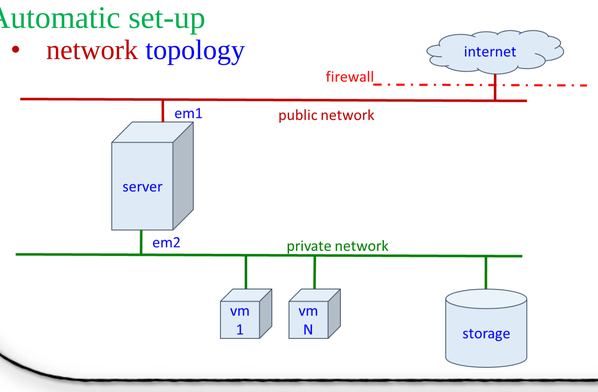
Main functions
- Check Task queue and start VMs
- Contextualize VMs to be WNs to the DIRAC WMS
- Pull jobs from central task queue
- Centrally monitor VM status
- VMs automatic shutdown according to jobs queues



Automatic set-up of cloud (micro-)infrastructures

Automatic set-up
- create kickstart file
- prepare customized ISO (CentOS-6.7-x86_64-netinstall.iso)
- make bootable usb drive
- install on server or client
- test the installation

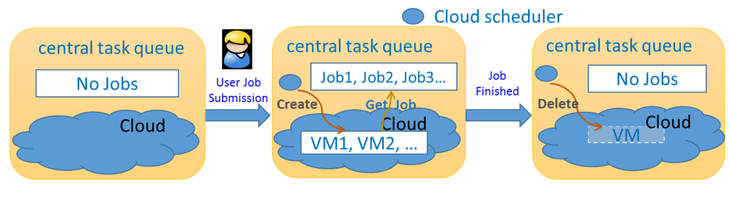
Automatic set-up
- network topology



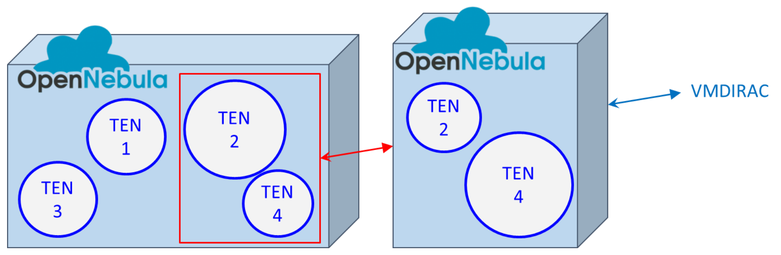
Ingredients for an 1h setup
- few parameters left to user input
  - network parameters
  - keyboard layout, ..
- automatic hypervisor setup
  - OpenNebula
  - squid proxy
  - rOCCI
- automatic WN setup
- full custom ISO for tenants
  - repository for VM/templates
  - QCOW2: dynamic increase
- minimal OS installation:
  - only required software installed
- contextualisation via CVMFS
- validation:
  - Sunstone
  - create VMs locally
  - run jobs locally
  - submitting jobs via VMDIRAC

Elasticity

On-demand usage
- elastic approach to cloud usage
- don't occupy resources before jobs are coming:
  - save money when you use commercial cloud
- VMDIRAC is one of the possible approaches allowing to use clouds elastically:
  - HTCondor + Cloud scheduler, elastiq, …
- central Task Queue and cloud scheduler are required



Modular approach
- exploiting easy automatic set-up of cloud (micro-)infrastructures
- two OpenNebula instances:
  - remote VMDIRAC drives the "private" OpenNebula
  - the "private" OpenNebula cloud bursts to the "public" cloud interface



"Private" OpenNebula
- stake-holder has full sys-man control
- can interface multiple tenants (remote VMDIRACs)
- groups all the stake-holder's tenants
- introduces one more layer in elasticity
- act as a simple tenant on the public OpenNebula

"Public" OpenNebula
- stake-holder
  - acts as simple tenant
  - has user-level control

A modular (almost) automatic set-up for elastic multi-tenants cloud (micro)infrastructures

Real life scenarios
- consortium of independent tenants
  - independent experiments (HEP) or applications (SME)
  - want to act as a single stake-holder in large cloud infrastructures
- stake-holders proxying theirs different tenants: experiments (HEP) or deparments (LE)
  - common resources procuring
  - want to decouples internal accounting/access to resources from large CI
- small groups (HEP) or applications (SME) with limited resources and cloud skills:
  - cloud micro-infrastructures exploiting automatic set-up
  - can interface with remote VMDIRACs or incoming cloud-bursting

"Public" Cloud Infrastructure (OpenNebula or OpenStack)
- is not aware of the specific interaction with remote VMDIRACs (limited to the "private" ONs)
- only interaction is cloud bursting with "private" OpenNebulas
- gives stake-holders only user access to the "public" clound infrastructure
- can still perform elastically with the different stake-holders
- if the "public" clound infrastructure is managed directly by the stake-holder:
  - can exploit automatic set-up of cloud (micro-)infrastructures
  - can receive clould-bursting from trusted remote OpenNebulas

"Private" Open NebulaCloud Infrastructure (OpenNebula or OpenStack)
- stake-holder has full control at sys-man level
- interfaces with (remote) VMDIRACs of the different tenants of the stake-holder
- has full control of the tenants' quotas in the "private" ON and hence on the "public" CI
- can perform elastically with the different tenants at the "private" ON level
- can elastically release resources on the "public" CI when not needed

CHEP2016
22nd International Conference on Computing in High Energy and Nuclear Physics
10th-14th October 2016 – San Francisco (USA)

The BESIIICGEM project has been funded by European Commission within the call H2020-MSCA-RISE-2014.