BESIII & BELLEII
Cloud Activities and R&D @ INFN-TO

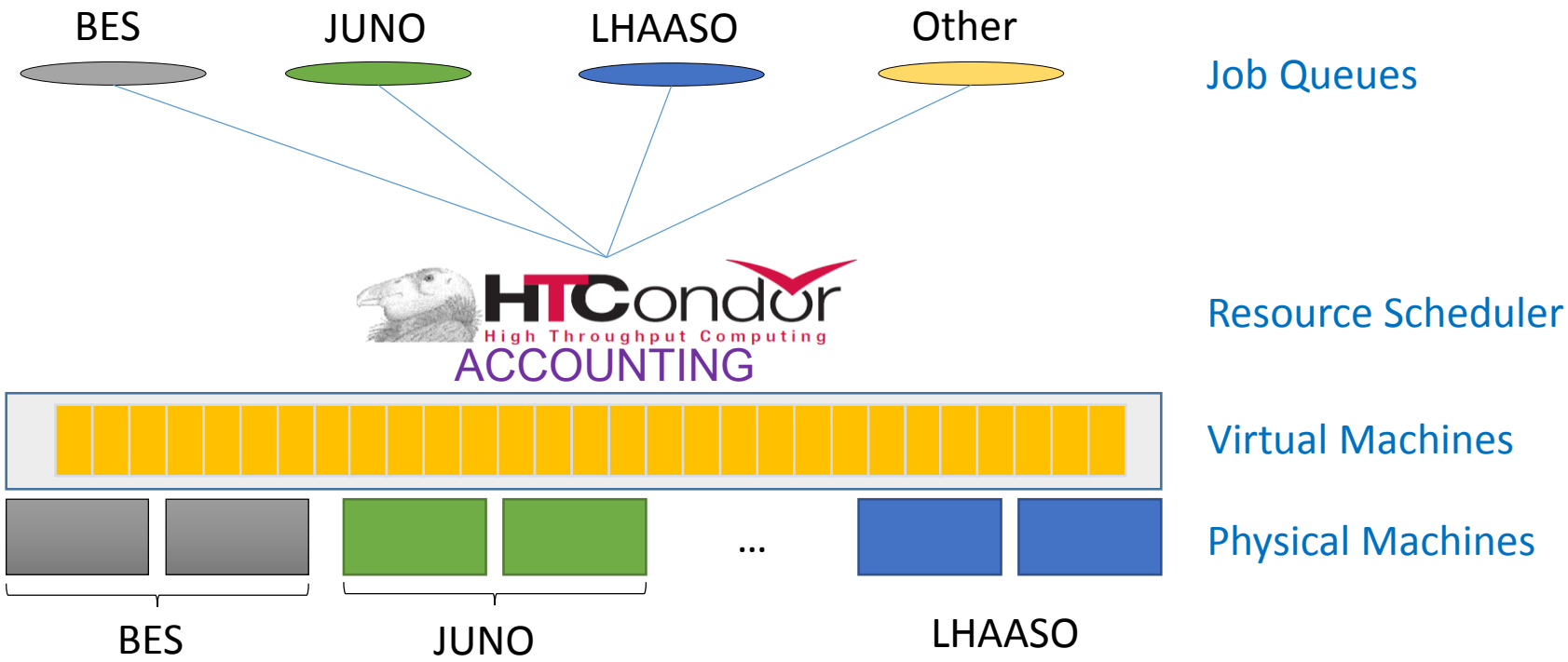
Marco Destefanis

University of Turin and INFN Turin

INFN-TO Cloud Mini-workshop
Torino, May 26th 2016

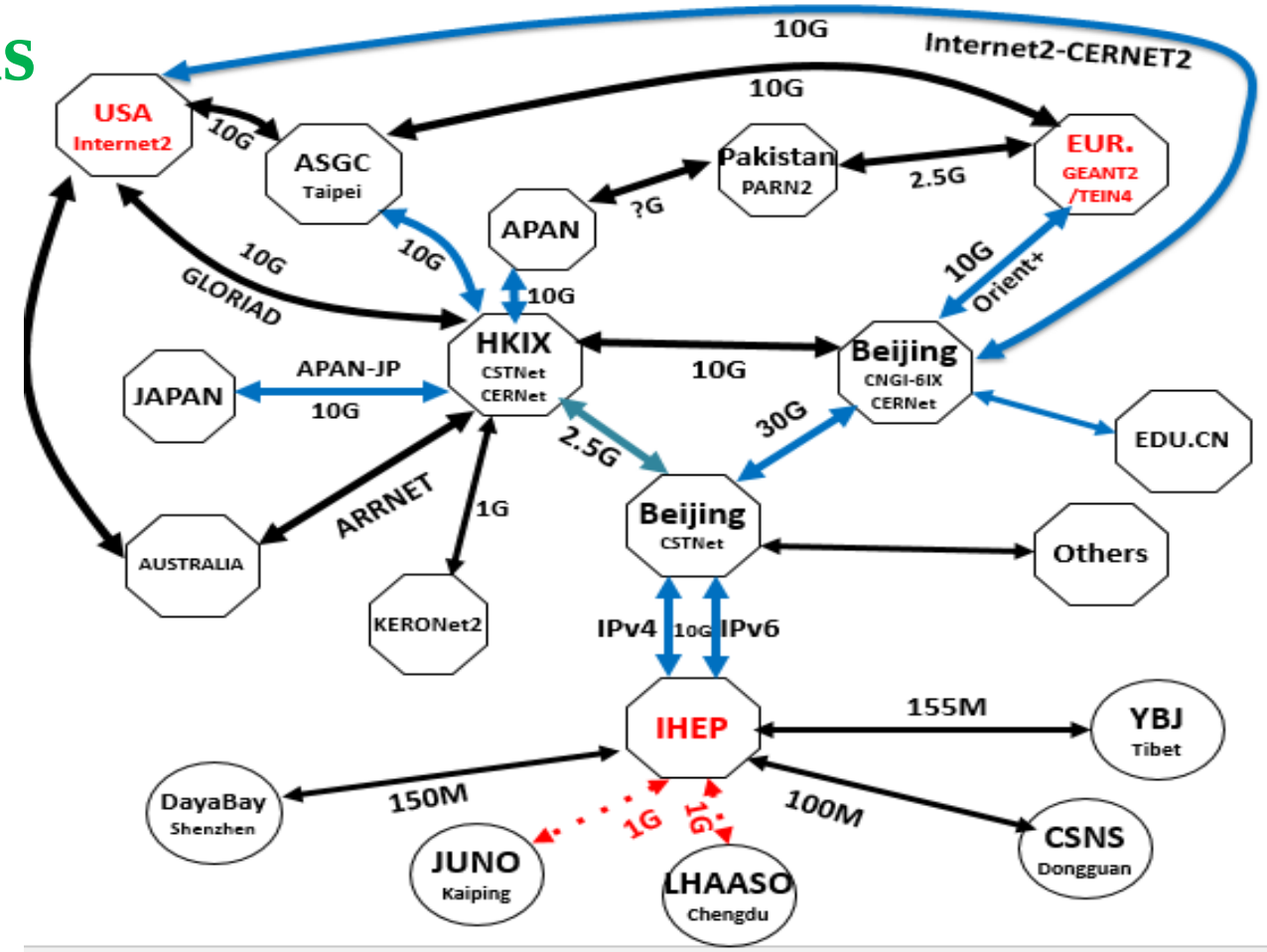
Cloud Computing @ IHEP: Future Architecture of Virtual Computing Cluster

- building virtual computing cluster
  - easy to share resources
  - improve resource efficiency
  - improve operational efficiency
- future architecture: four layers
  - 1st layer: Physical machines
    - bought and owned by
    - different experiments
  - 2nd layer: Virtual machines
    - shared resource pools,
    - not belong to any experiments
  - 3rd layer: Resource scheduler
    - dynamically allocate resources to different experiments depending on the task list
    - resource allocation policies to balance the resource sharing and physical machine invest
  - 4th layer: job queues
    - different job queues for end users of different experiments
    - same way to use as traditional cluster



IHEP links

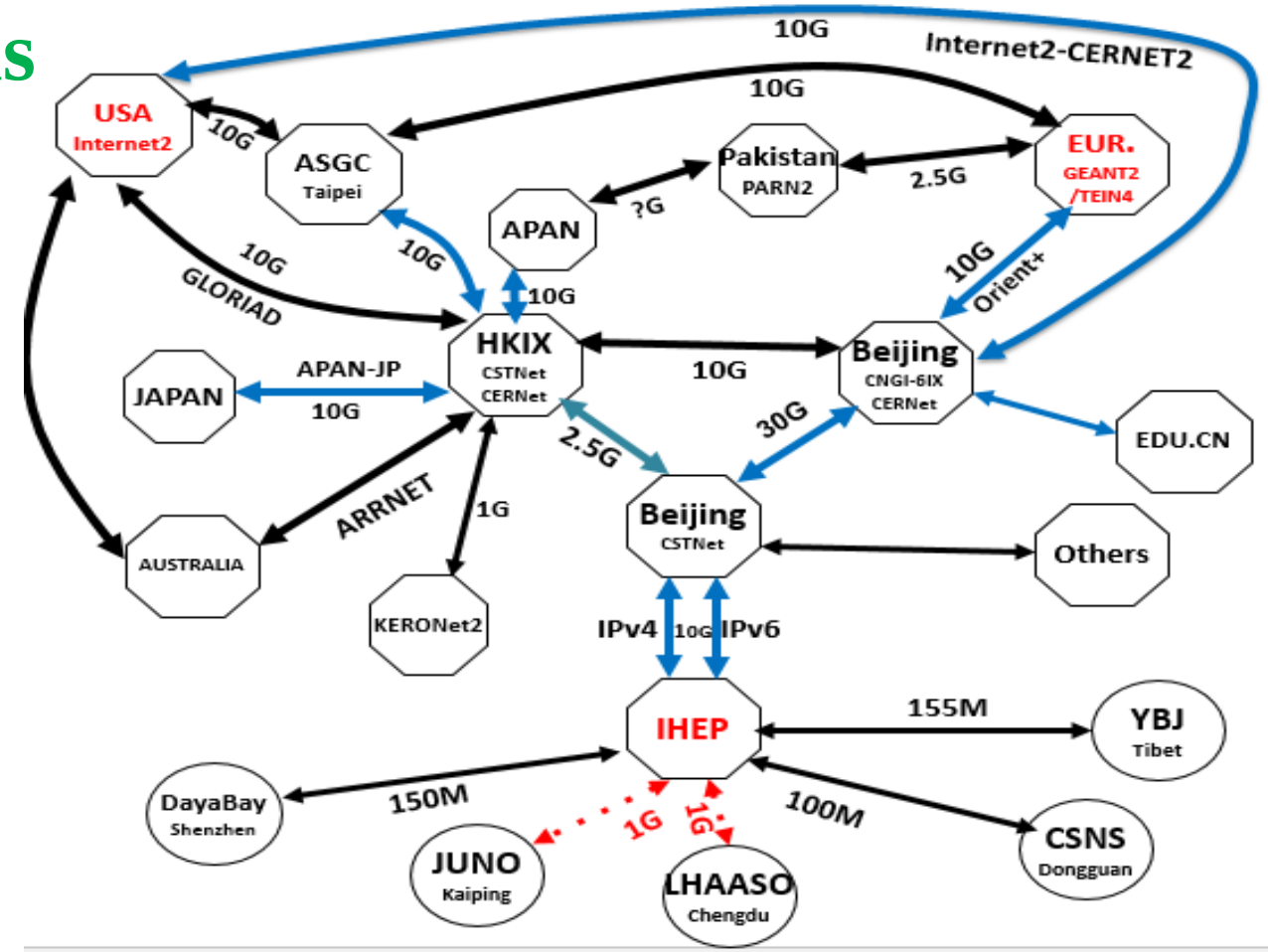
- **IHEP internet Connections**
  - IHEP-EUR.: 10Gbps
  - IHEP-USA: 10Gbps
  - IHEP-Asia: 2.5Gbps
  - IHEP-Univ: 10Gbps



- **PerfSONAR@IHEP**
  - Bandwidth: Perfsonar.ihep.ac.cn
  - Latency: Perfsonar2.ihep.ac.cn

IHEP links

- **IHEP internet Connections**
  - IHEP-EUR.: 10Gbps
  - IHEP-USA: 10Gbps
  - IHEP-Asia: 2.5Gbps
  - IHEP-Univ: 10Gbps



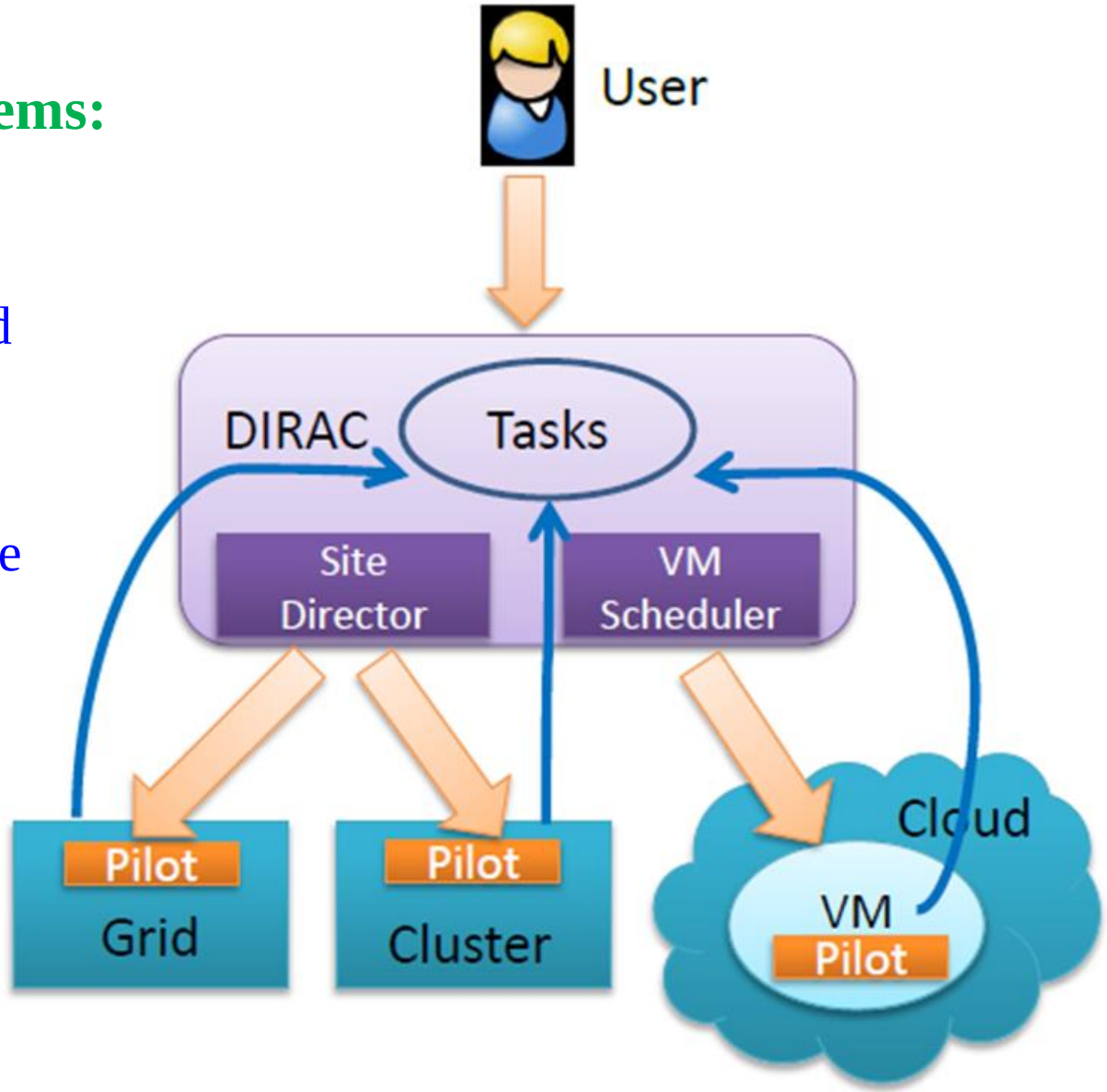
- **PerfSONAR@IHEP**
  - Bandwidth: Perfsonar.ihep.ac.cn
  - Latency: Perfsonar2.ihep.ac.cn

A network suited for an effective BESIII Distributed Computing!

BESDIRAC elastic workload management

- **Global integration to different echosystems:**
  - job scheduling scheme remains unchanges
  - instead of site Director for cluster and grid, VM scheduler is introduced to support cloud

- **Elastic workflow:**
  - start new virtual machine with one CPU core when there are waiting jobs
  - one job scheduled on each virtual machine at the same time
  - delete the virtual machine after no more jobs for a certain period of time

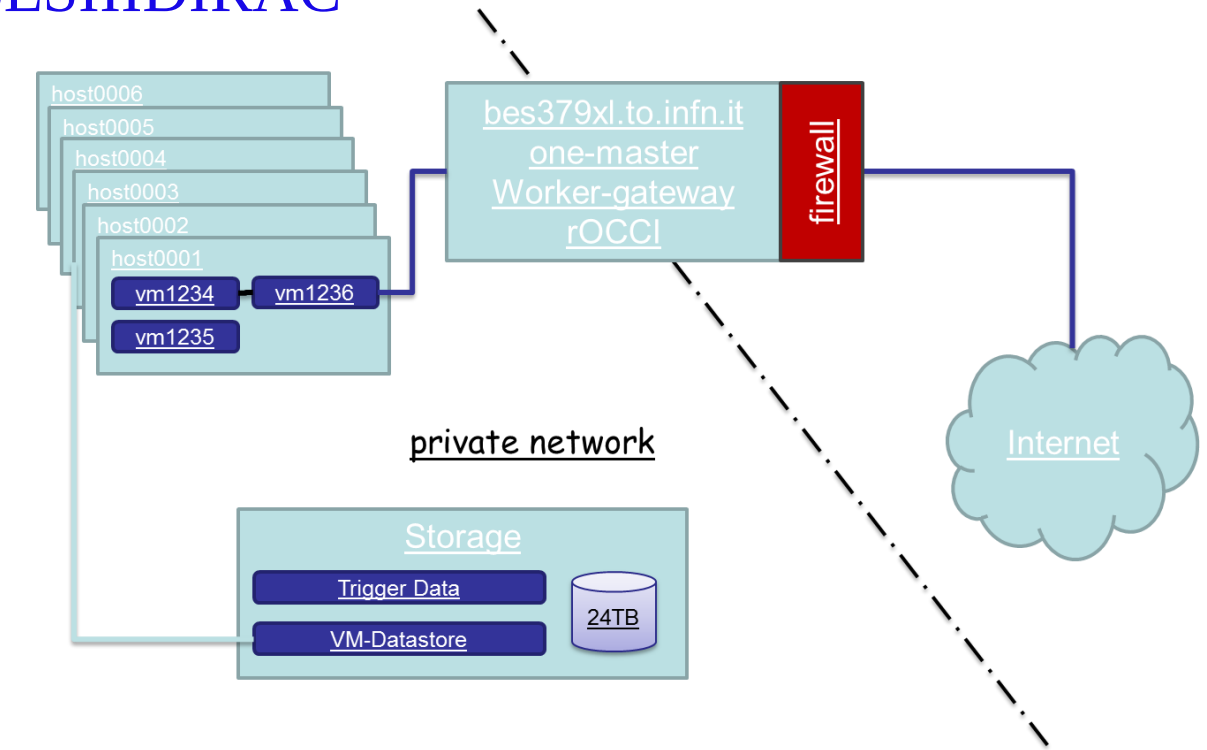


The INFN virtualised on cloud BESIII Grid Tier-2: GRID.INFN-Torino.it

- **cloud infrastructure at INFN-TO Computing Center:**
  - cloud infrastructure optimised for scientific computing
  - virtualised: VMs, farms, full Tier-2 Grid infrastructures, LANs
  - based on OpenNebula / CVMFS / Squid / BESIIIDIRAC
- **cached contextualization:**
  - squid + CVMFS Stratum 0 or 1
- **BESIII activities:**
  - 2kHS06 (~ 200 cores) and 60 TB net reserved for BESIII (Random Trigger Data ~ 25 TB)
  - shared access to 0.7kHS06
- **INFN provides to BESIII:**
  - fully transparent Tier-2 Grid Infrastructure, accessible by IHEP
  - direct submission to CE, contestualisation via CVMFS
  - job submission in BESIIIDIRAC included, from INFN and IHEP
  - part of BESIII mass productions

The INFN-TO Cloud Lab, a Micro Cloud Infrastructure devoted to R&D: CLOUD.Torino.it

- **R&D cloud infrastructure at INFN-TO Computing Center:**
  - cloud infrastructure for R&D
  - virtualised: VMs, farms, full Tier-2 Grid infrastructures, LANs
  - based on OpenNebula / CVMFS / Squid / BESIIIDIRAC
- **cached contextualization:**
  - squid + CVMFS: CERN Stratum 0
- **BESIII activities:**
  - 1.2kHS06 (128 cores) and 19 TB net reserved for BESIII but not exported
- **INFN provides to BESIII:**
  - a complete test bench for R&D on cloud technologies
  - able to cope with all the servers/clients/agents of the production cloud, and more



BESIII Distributed Computing

#	Site Name	CPU Cores	Storage	Status		#	Site Name	CPU Cores	Storage	Status
1	CLOUD.IHEP.cn	210	214 TB	Active		8	GRID.JINR.ru	100 ~ 200	30 TB	Active
2	CLUSTER.UCAS.cn	152		Down		9	GRID.INFN-Torino.it	200	60 TB	Active
3	CLUSTER.USTC.cn	200 ~ 600	24 TB	Down		10	CLOUD.TORINO.it	128		Active
4	CLUSTER.PKU.cn	100		Down		11	CLUSTER.SDU.cn	100		Testing
5	CLUSTER.WHU.cn	120 ~ 300	39 TB	Active		12	CLUSTER.BUAA.cn	100		Testing
6	CLUSTER.UMN.us	768	50 TB	Active		13	GRID.INFN-ReCas.it	50	30 TB	Active
7	CLUSTER.SJTU.cn	100		Active		14	CLOUD.CNIC.cn	50	50 TB	Active

Total resources:

- ~ 1700 CPU cores
- ~ 500 TB storage

INFN contributions to BESIII:

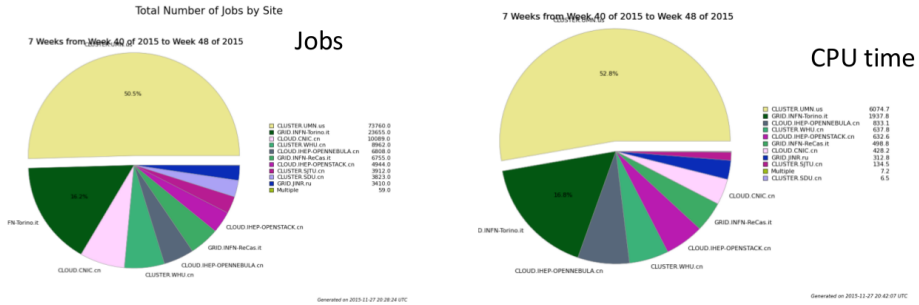
Grid Infrastructures: 250 cores & 90 TB

Cloud Lab

BESIII Distributed Computing: significant INFN contribution

Site Contributions

- UMN site is still the largest contributor in last three month.
  - More than 75K jobs executed at UMN
  - 50.5% of the jobs, and 52.8% of the CPU time
- INFN-Torino becomes the second largest contributor
- UMN site is even the largest contributor in last two years
  - Stable, fast response and good communications
  - Thank Hajime Muramatsu and UMN local computing group for good supports



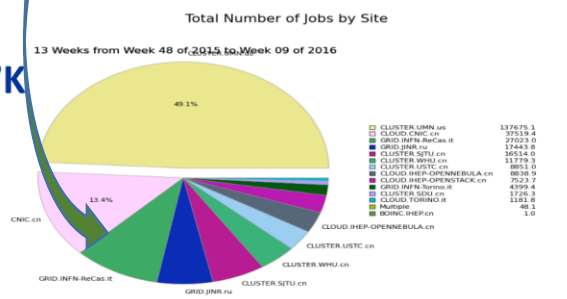
4

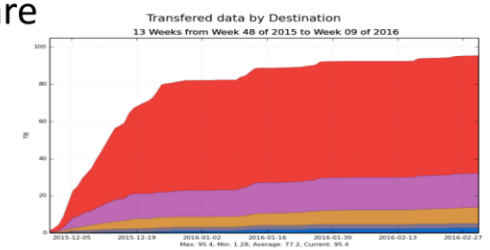
Prof. Xiaomei ZHANG, BESIII CM 12/2015

GRID.INFN-ReCas.it

BESIII Distributed Computing

- In the last three months, about **137K** BESIII jobs have been run on the platform
  - 12 sites join the production
  - About half of contribution from the UMN site
- Total data exchange among sites are about **95.4TB**, including
  - Access of random trigger data
  - Write back job outputs
  - Transfer random trigger data
- Welcome to use distributed computing and give your feedback!





BESIII Workshop 2016 12nd March, 2016 15/22

Prof. QiuLing YAO, BESIII PCW 03/2016

Providing small and medium sites access to cloud technologies

And what if…

instead of exploiting commercial (i.e. no control or LTS!) at high prices

we make life easier for those academic sites

who wants to deploy CI able to cope with

VMDIRAC & BESDIRAC?

Ingredients:

- OpenNebula
- rOCCI
- kickstart
- squid

B3CT:
BESIII Cloud Toy v0.1

Providing small and medium sites access to cloud technologies

And what if…

instead of exploiting commercial (i.e. no control or LTS!) at high prices

we make life easier for those academic sites

who wants to deploy CI able to cope with

VMDIRAC & BESDIRAC?

Ingredients:

- OpenNebula
- rOCCI
- kickstart
- squid

Easy Cloud Infrastructure Setup

Goal: deploy an OpenNebula hypervisor
minimizing the user interaction
during the installation process

Mean: server installation via usb key

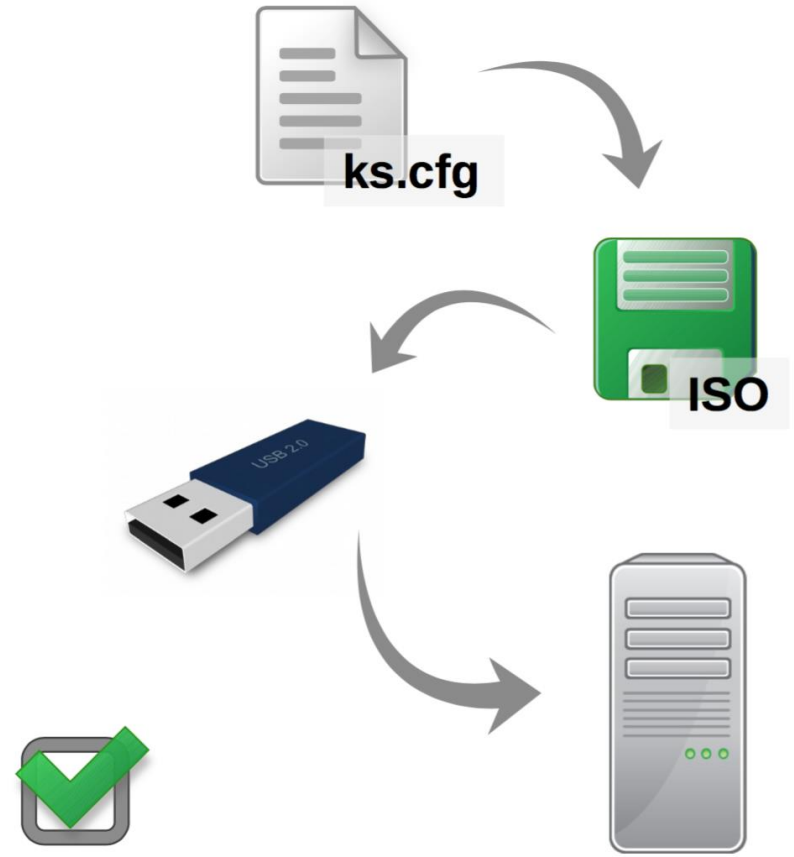
B3CT tested on:

Machine	Dell Server
CPU	2 x Intel(R) Xeon(R) E5-2650 v3 @ 2.30GHz
Cores	20 physical 40 hyper treading
RAM (GB)	160 GB

B3CT: a step by step receipt

Can/must be perfomed at every major upgrade of OpenNebula/rOCCI/squid or BESDIRAC

- create kickstart file
- prepare customized ISO
- make bootable usb key
- install on server
- check everything works



B3CT and BESIIIDIRAC

- BESDIRAC deploys the "VM director" to a B3CT CI
- a repository provides images and templates of the VMs for the B3CT CI
- a proper VMs in instantiated if necessary and a pilot job is executed, or…
- an available VM receives from OpenNebula a pilot job
- any VM at instantiation contextualizes to BESIII via CVMFS
- a squid server provides a fast local cashing during CVMFS contextualization
- the pilot job pulls jobs from the stack
- any unused VM is killed freeing resources

B3CT and BESIIIDIRAC: advantages

Little and medium size sites:

- easier access to cloud technologies and reduced manpower for CI deployment
- reduced manpower for Hypervisor and server upgrades
- easy retrieval of images and templates optimized for BESIII SW
- no manpower for BESIII SW updates

IHEP and BESIII SW management:

- standardization of the sites participating to the BESIII Distributed Computing (DSC)
- VM-pilots can be executed by a centralized BESDIRAC console at IHEP:

  less manpower, larger control
- localized knowledge can provide images and templates to the whole BESIII DSC

BESIII/BELLEII OpenNebula/VMDIRAC elasticity

Intra-experiment elasticity:

- operational through BESDIRAC (i.e. VMDIRAC) for BESIII
- in test stage through BELLEII VMDIRAC for BELLEII

Inter-experiment elasticity at Infrastructure level:

- advanced design stage exploiting infrastructure-side tools interfaced **dynamically** with the different VMDIRAC's of the stakeholder
- VM-pilots can instantiate new VMs profiting of **dynamic quotas**
- VMDIRAC kills **unnecessary VMs**
- infrastructure-side tools **update dynamically** the different **VMDIRAC quotas**

BESIIICGEM Outreach: IHEP-INFN Joint Doctoral School on Cloud Computing

Joint IHEP-INFN Doctoral School: Sep. 7th – 11th, 2015

- funded by BESIIICGEM, CCAST and IHEP
- main audiences are Doctoral Students in the High Energy or Nuclear Physics fields or in the IT and Computing fields
- plays an important role of pushing forward cloud technologies within BESIII and other HEP Collaborations within P.R.C.
- highly valued by EU H2020-RISE PO and referee

Joint IHEP-INFN Doctoral School: Jul. 18th – 22nd, 2016

- in Shandong University, Jinan

Thank you!

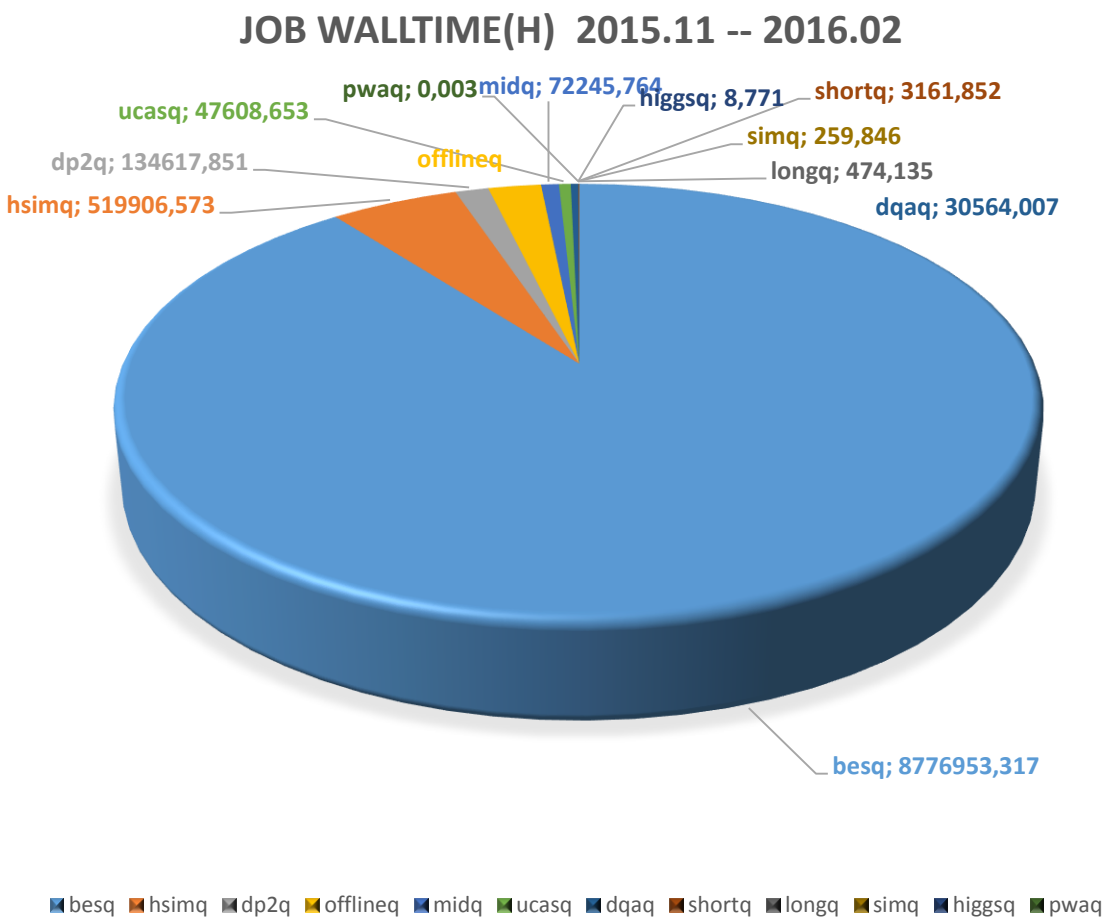
IHEP Spares

IHEP cluster

New Resources Since Sep 2015

- **86** blade servers have been added to the BES batch system
  - Lenovo Flex System x240 M5
  - CPU E5-2680 v3
  - Total CPU cores is **2064**
- 368 slow CPU cores have been retired
- computing power has increased by **50%**
  - HEPSPEC06 Before: 75.5kHS06 Now: 116.5kHS06

BESIII job statistics @ IHEP

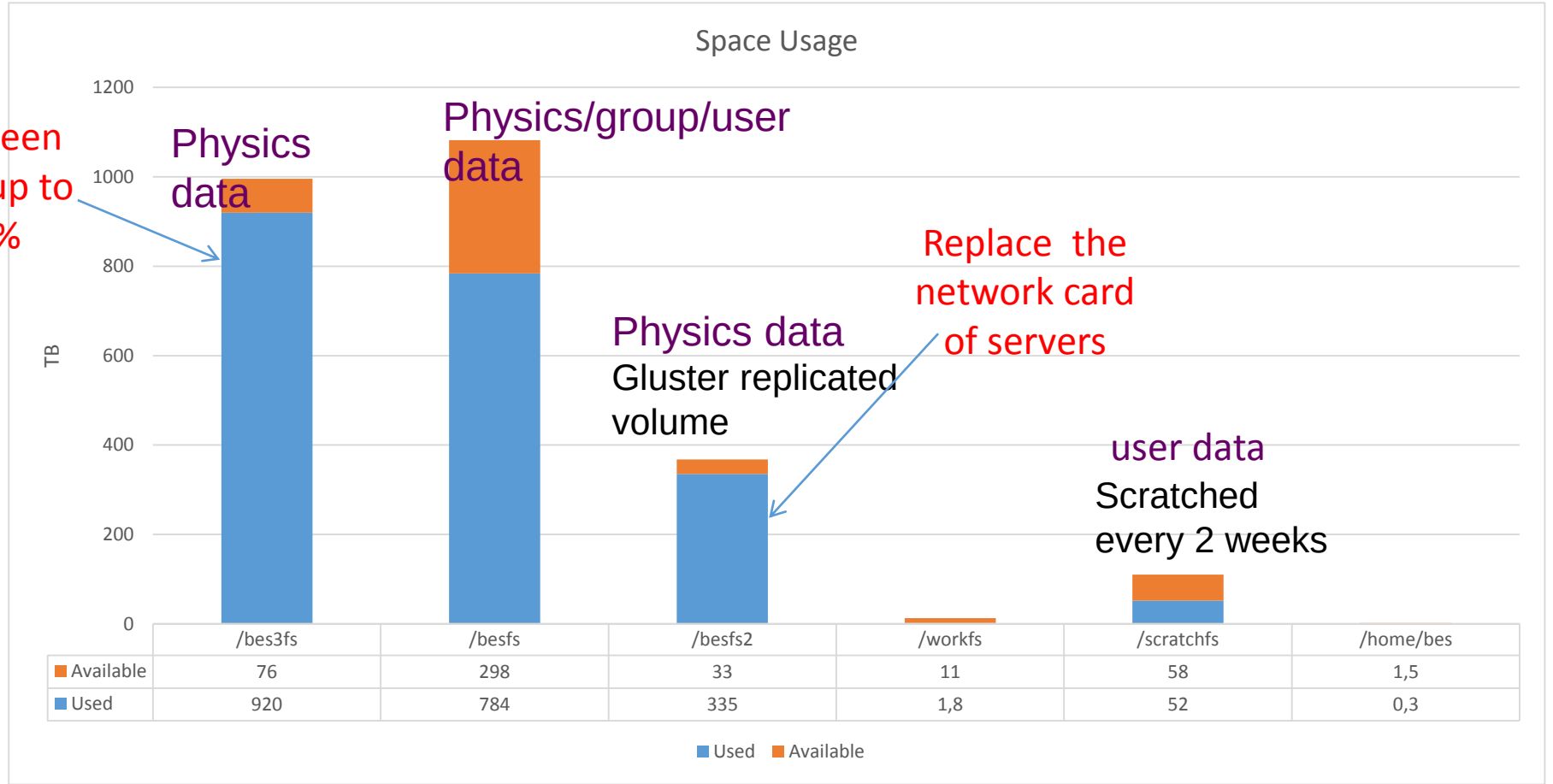


- BESIII group has submitted **1,722,333** jobs
- compared to 2014, walltime in 2015 increases by **12.2%**
- besq has comsumed **87%** of the total walltime and **79%** of the total jobs
- dp2q has been used **less** than before (thanks to DSC!)

Queue name	Aim for	priority
dp2q	Simulation & Reconstruction	High
besq	analysis	Middle
midq	Middle job	Low
longq	Long job	Low
pwaq	Pwa job	Low
dqaq	DQA jobs	Middle
hsimq	Higgs jobs	Very low

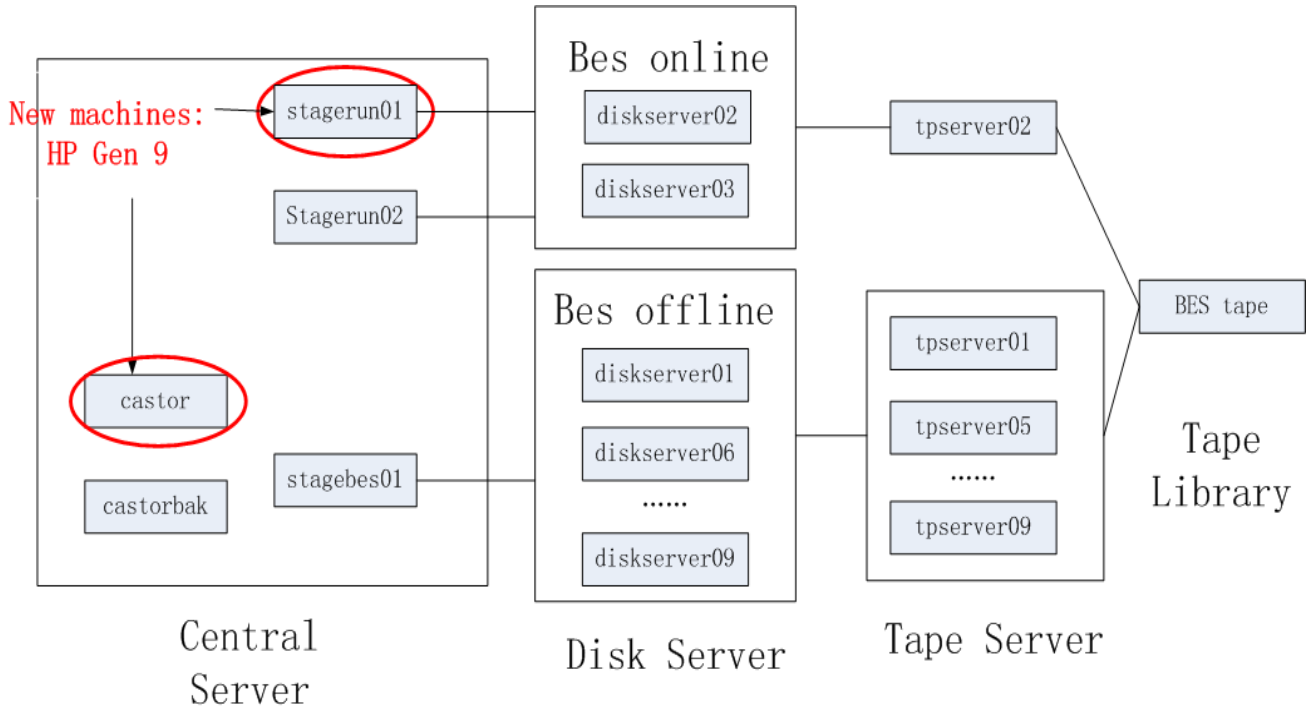
BESIII storage @ IHEP

- **4 dedicated** file systems for BESIII, **2 shared** files systems
- **2.5 PB** total space, **2PB** has been used, **100M** files has been stored.



BESIII hierarchical storage @ IHEP

- **2** data storage mode: disk array and tape library(IBM 3584)
- BES online data has been stored **in real-time**
- raw data is stored in **two copies on tapes**, with one copy is kept in the main building
- replaced the central server
  - castor,stagerun01
  - backup server:castorbak,stageun02
  - synchronize the databases between castor



	BES online	BES offline	Ratio
File count	79802	1258046	20%
Tape usage （TB）	823.484	1526.446	46%

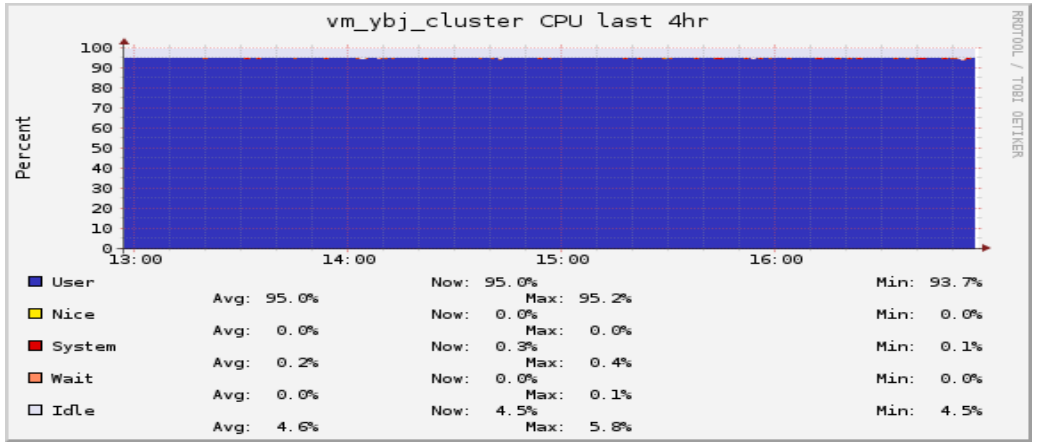
BESIII local batch system @ IHEP

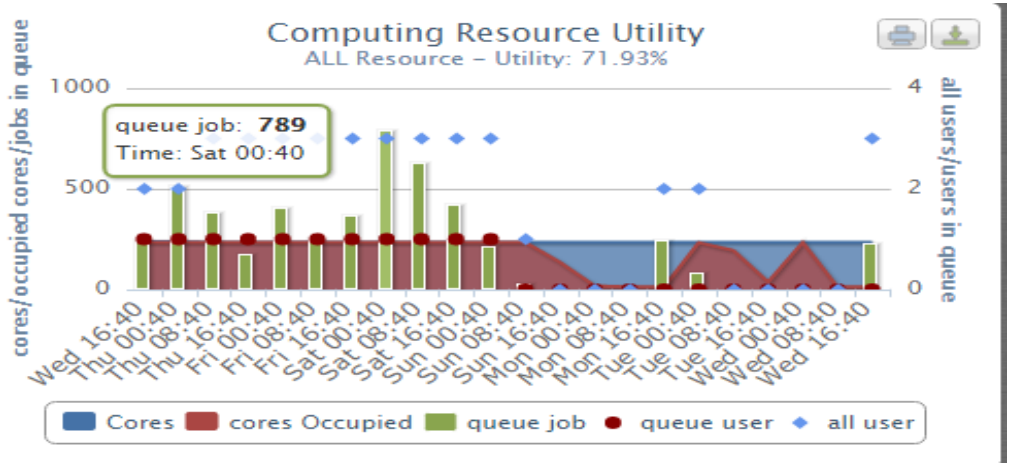
- BES Batch system migration plan
  - HTCondor, a new job scheduling service
    - migrate part of computing nodes from PBS to HTCondor during the summer maintenance period.
    - migrate all nodes to HTCondor eventually (if everything is smooth)

- Plan of HPC (High Performance Computing)
  - a new heterogeneous hardware platform : CPU, Intel Xeon Phi, GPU
  - parallel programming supports : MPI, OpenMP, CUDA, OpenCL …
  - potential usage cases : simulation, partial wave analysis …
  - evaluation is underway.
  - network Architecture & technologies
    - InfiniBand network for HPC testbed will be setup soon

Cloud Computing @ IHEP: IHEPCloud

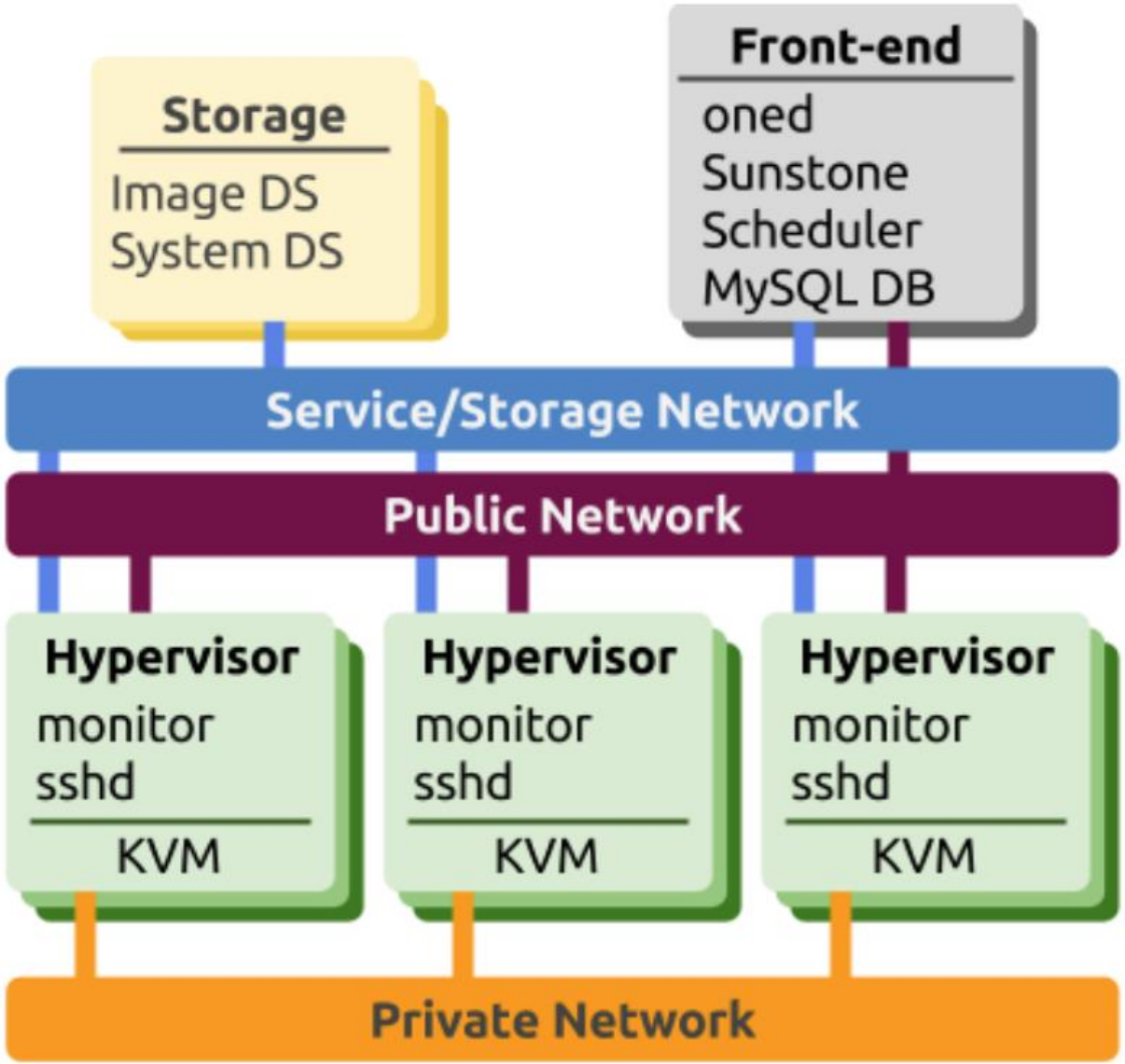
- based on Openstack Kilo
- **30** physical machines, **720** CPU cores
- job queues managed by HTCondor
- file systems are mounted the **same** way as local batch system
  - /bes3fs, /besfs, /workfs, …
- supports LHAASO and JUNO currently
  - start Virtual machines in advance by system administrator
  - dynamic scheduling system under development
- good efficiency for LHAASO simulation jobs
  - CPU efficiency: **99.4%**
    [Efficiency = CpuTime / Walltime]
  - stable run for more than 2 months
- BESIII jobs are forwarded to IHEPCloud by PBS
  - more testing to be done
  - to be integrated in the future





OpenNebula Spares

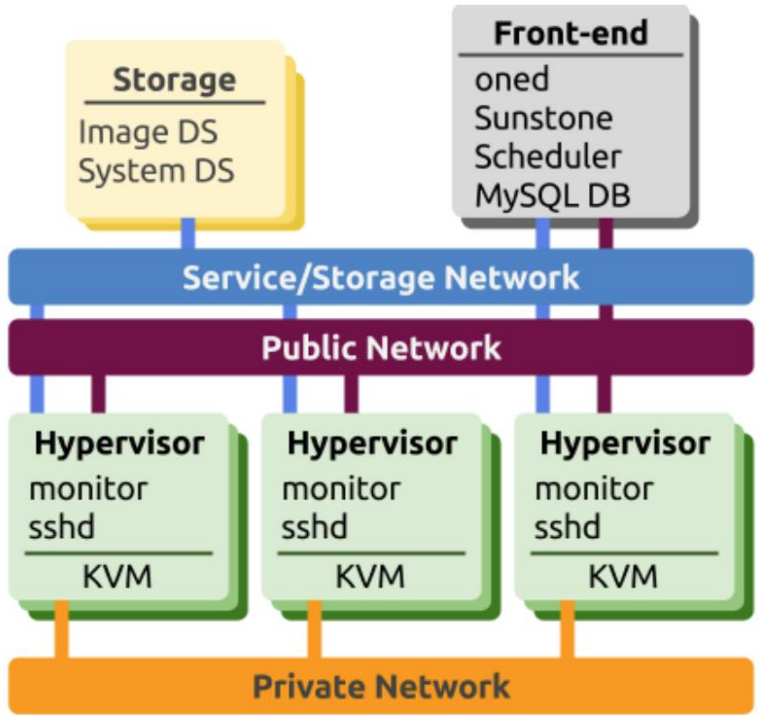
OpenNebula: Open Cloud Reference Architecture



OpenNebula: physical hosts

Servers that will host the Virtual Machines:

- often called “Hypervisors” (like the software)
- KVM
  (OpenNebula supports also vCenter and Xen)
- monitoring daemons
- sshd for system connection



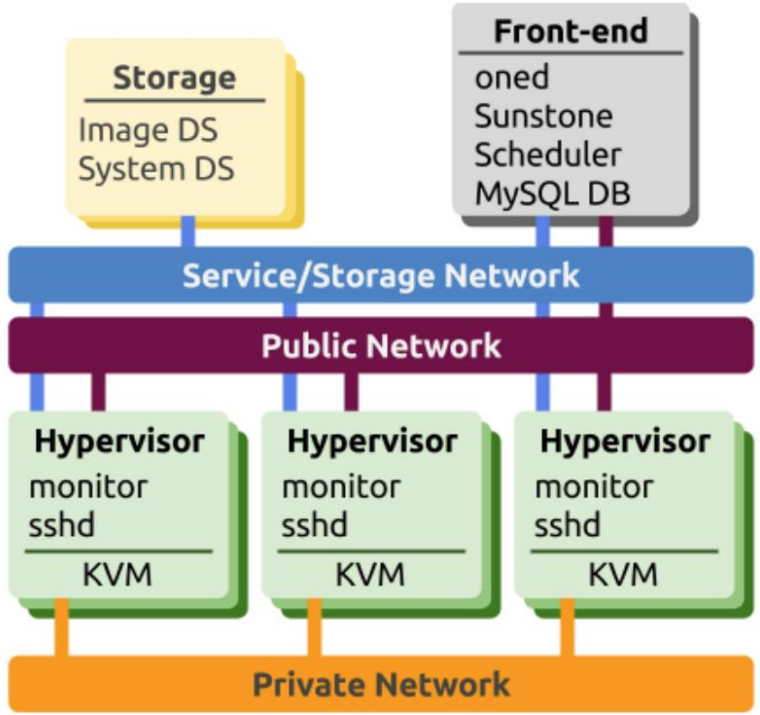
OpenNebula: networks

Used by OpenNebula and the infrastructure:

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

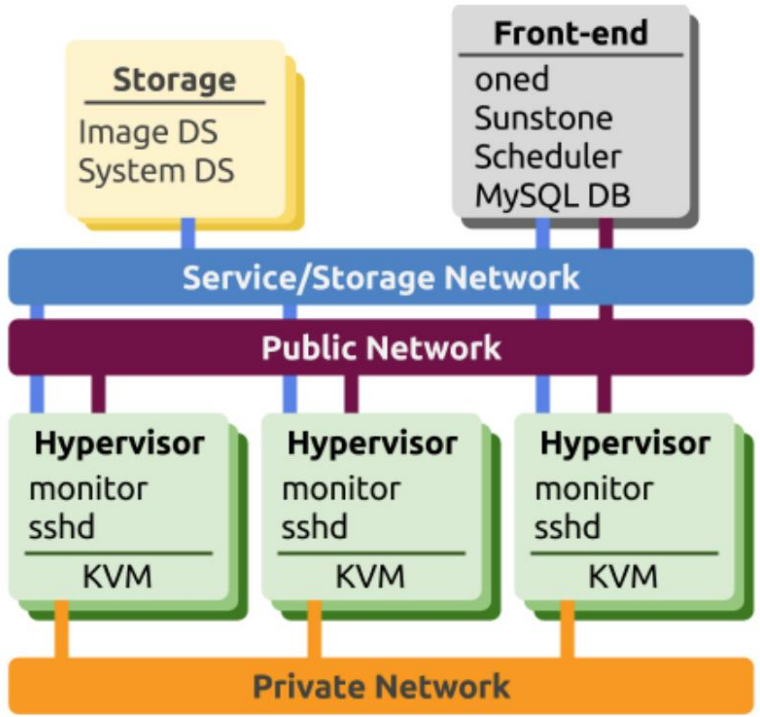
Used by the Virtual Machines:

- Private Network:
  - private IPs
  - intra-cloud communications
- Public Network:
  - public IPs
  - incoming connectivity to VMs



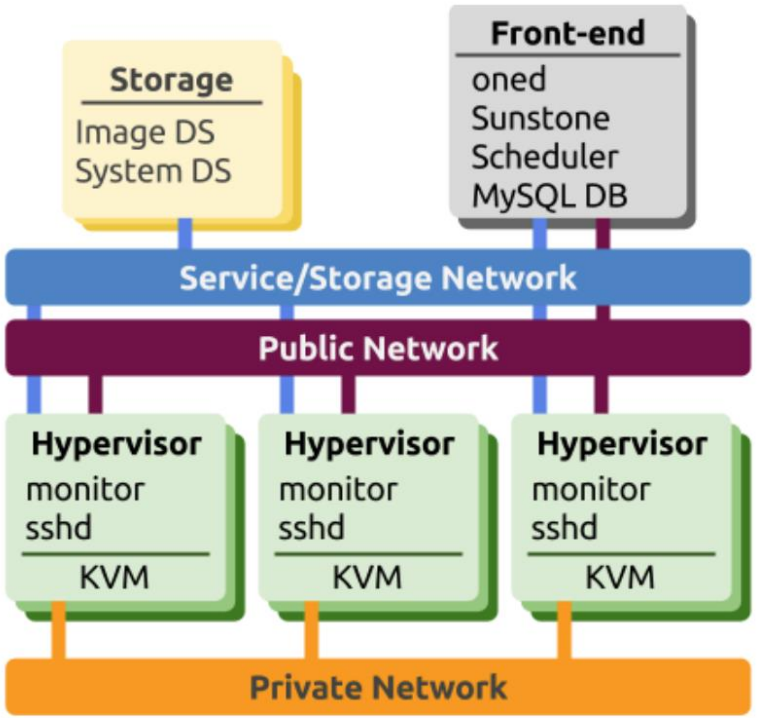
OpenNebula: storage

- 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"



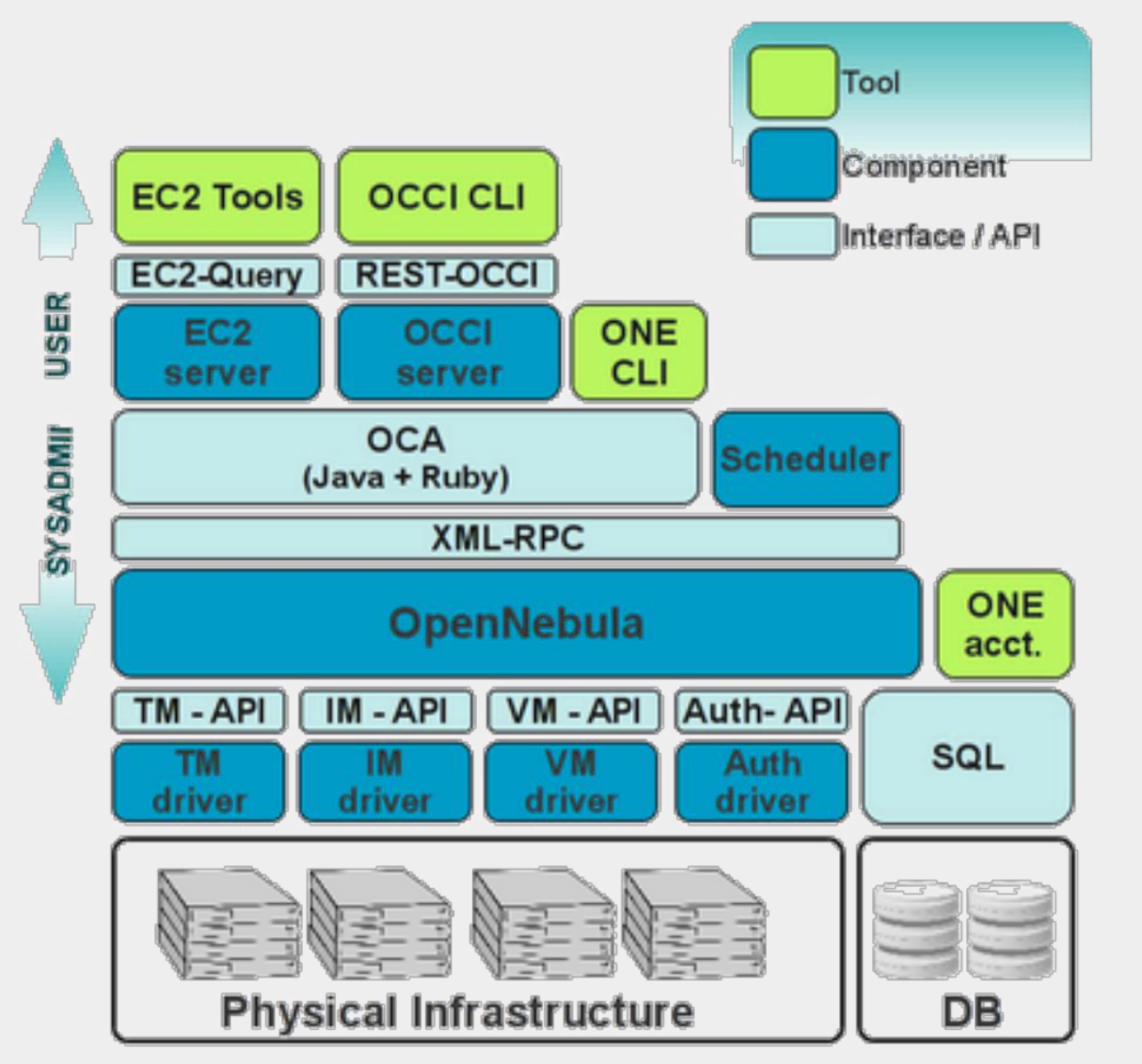
OpenNebula: the control node

- 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,…)

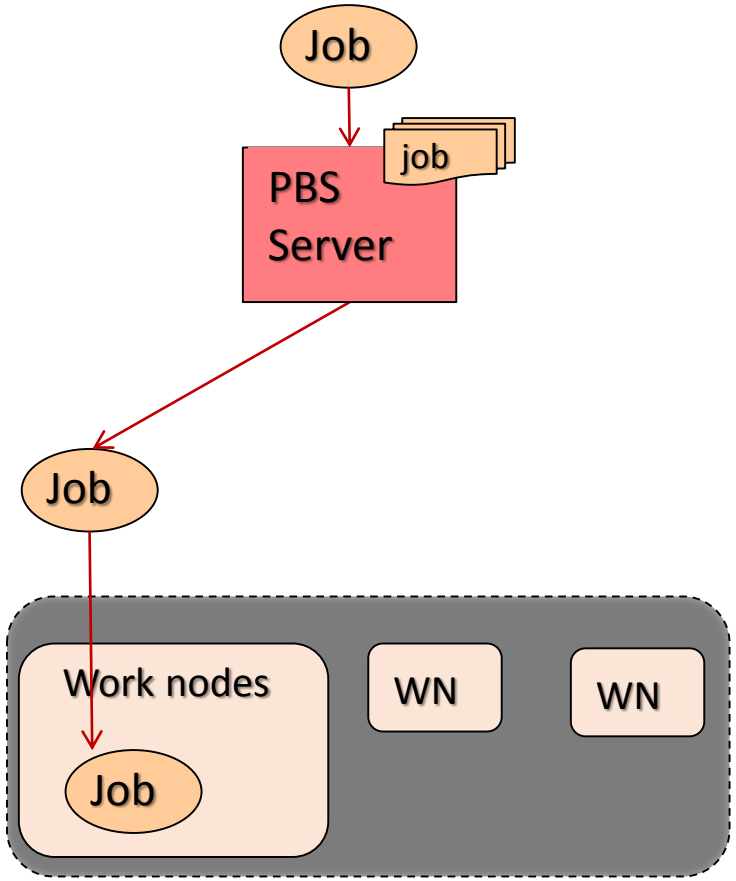
OpenNebula: internal architecture



DIRAC Spares

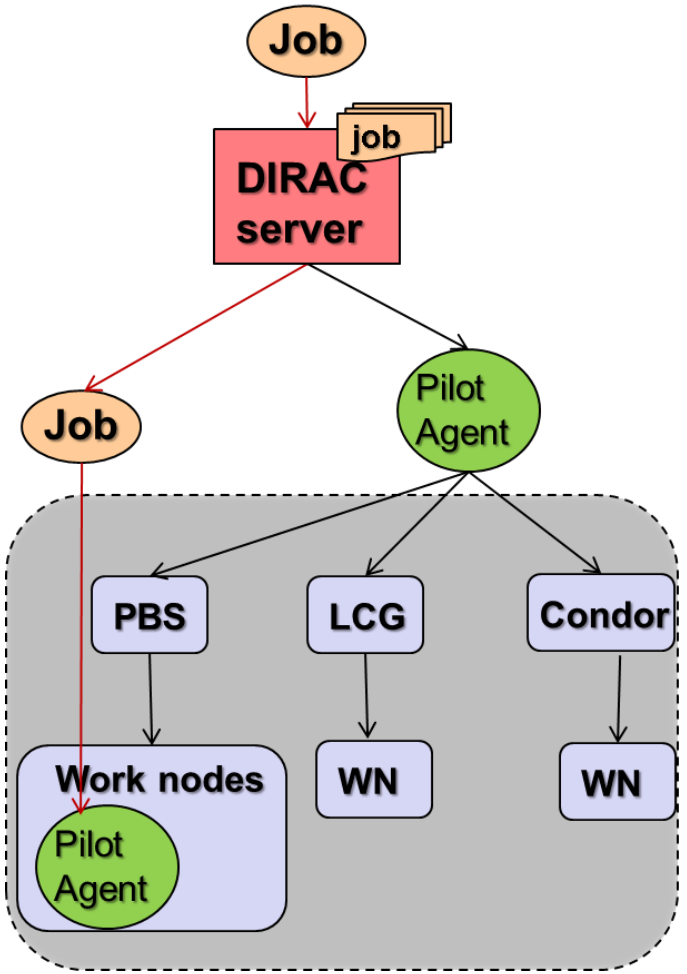
DIRAC push scheduling

- Two common ways to schedule jobs to resources
  - Push scheduling
  - Pull scheduling
- Push scheduling on clusters
  - User jobs is submitted to the local scheduler
  - Jobs are put into queues
  - Be arranged to WNs directly



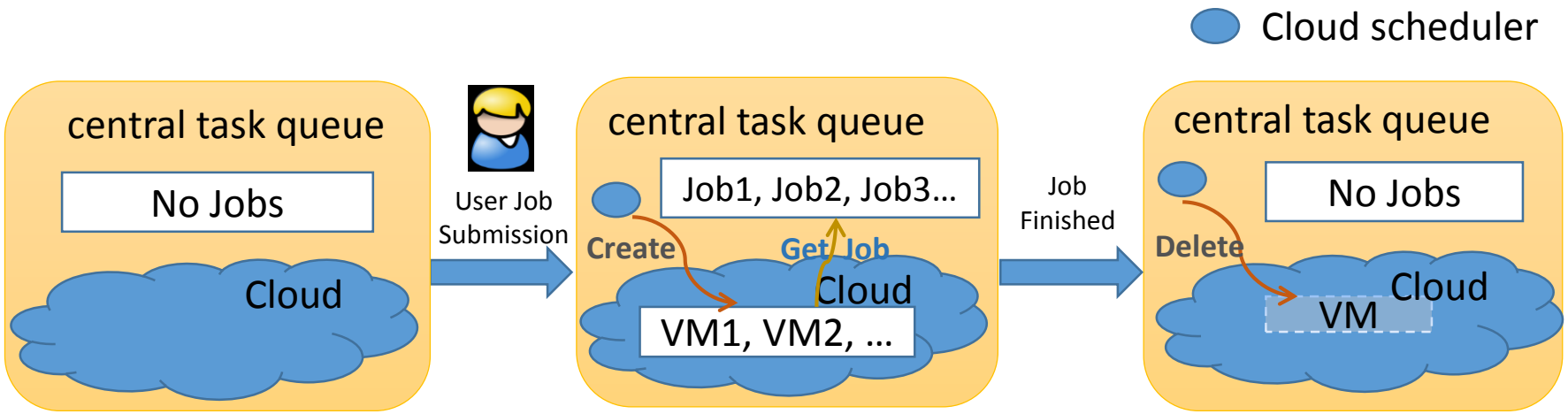
DIRAC pull scheduling

- Pull scheduling with pilot paradigm on DIRAC
  - Instead of send use jobs to resources directly
  - Pilot jobs are sent to resource brokers (CE, PBS…) as normal jobs
  - Pilot jobs start job agents
  - Job agents do
    - occupy a resource
    - set up environment
    - pull jobs from central queue
- Advantages
  - Avoid failure of user jobs because of hardware problem
  - Easy to fit in different resource environment



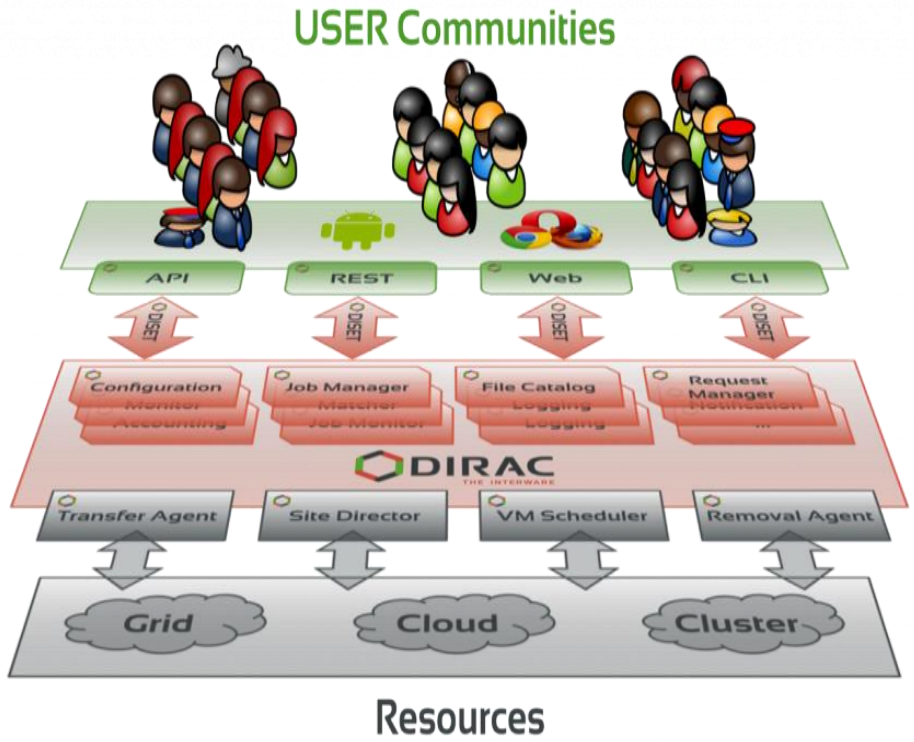
Elastic cloud

- On-demand usage
  - Elastic way to use cloud
  - Don’t occupy resources before jobs are coming
    - Save money when you use commercial cloud
  - VMDIRAC is one of the way allowing to use clouds elastically
    - HTCondor + Cloud scheduler, elastiq
  - Need central task queue and cloud scheduler



DIRAC

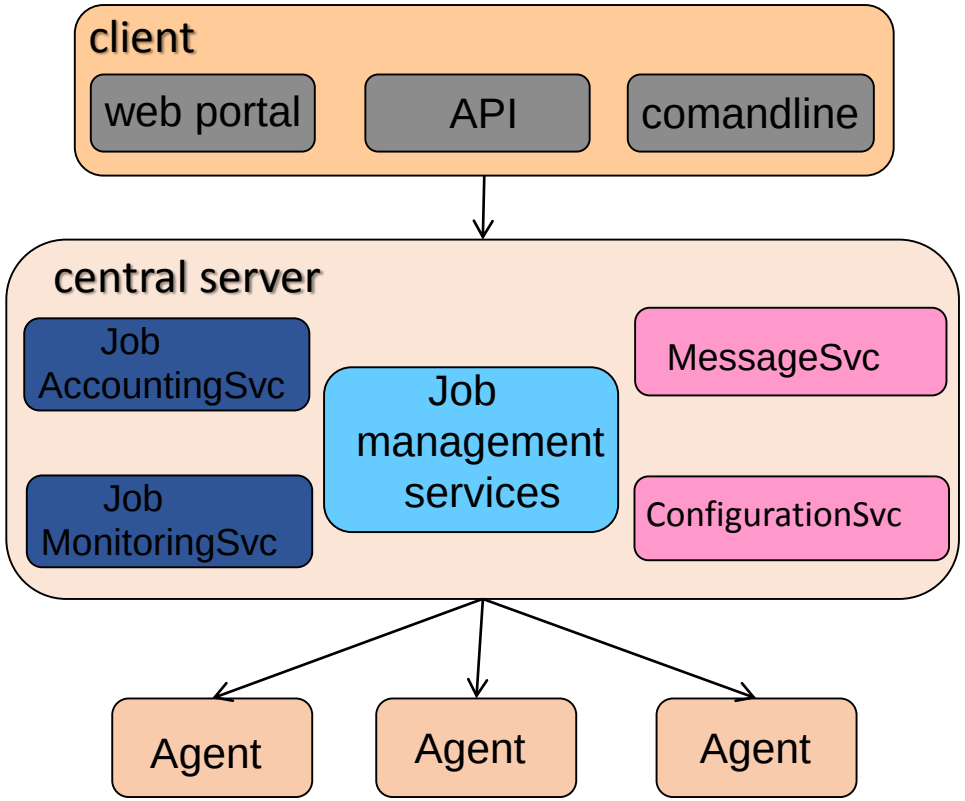
- DIRAC allows to interconnect computing resources of different types as a interware
  - Grid
  - Standalone Cluster
  - Desktop grid
  - Cloud



DIRAC systems

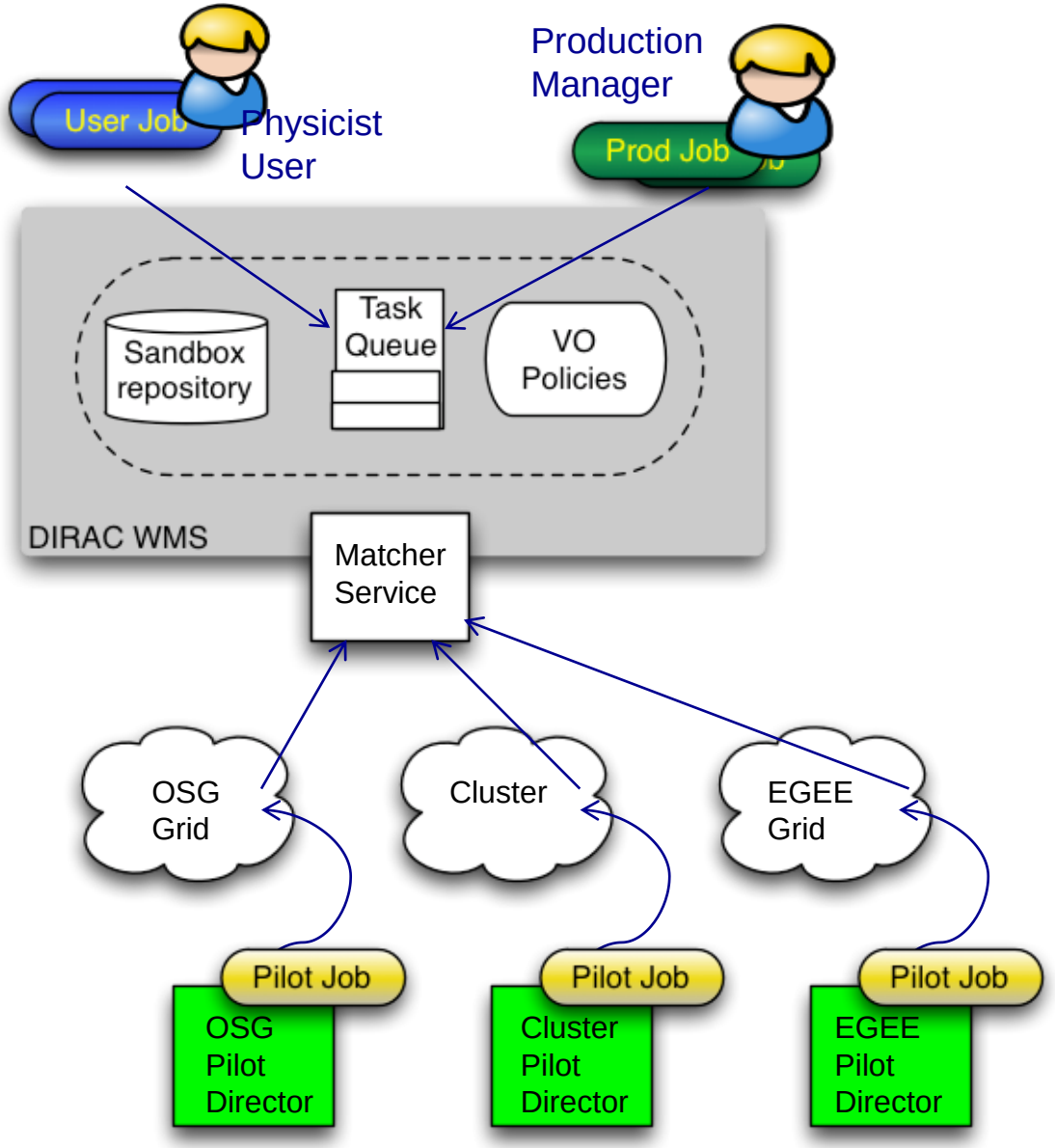
- VMDIRAC is one of DIRAC systems
  - Workload management, Data management….

- Each system consist of similar components:
  - services: passive components, permanently running, waiting for queries or requests
  - agents: light and active components which run as independent processes to fulfill one or several system functions
  - clients
  - databases



DIRAC workload management

- DIRAC is like a big cluster system over WAN
- Central task queue
  - User jobs are put into the task Queue
  - Job priorities are controlled with VO policies
- Pilot director
  - Connect with resource broker and submit proper pilots
  - Deal with heterogeneous resources
    - Every resource type need a pilot director
- Match service
  - Cooperate with pilot, Match proper user jobs to resources



INFN-TO Spares

INFN-TO cloud infrastructure - specifics

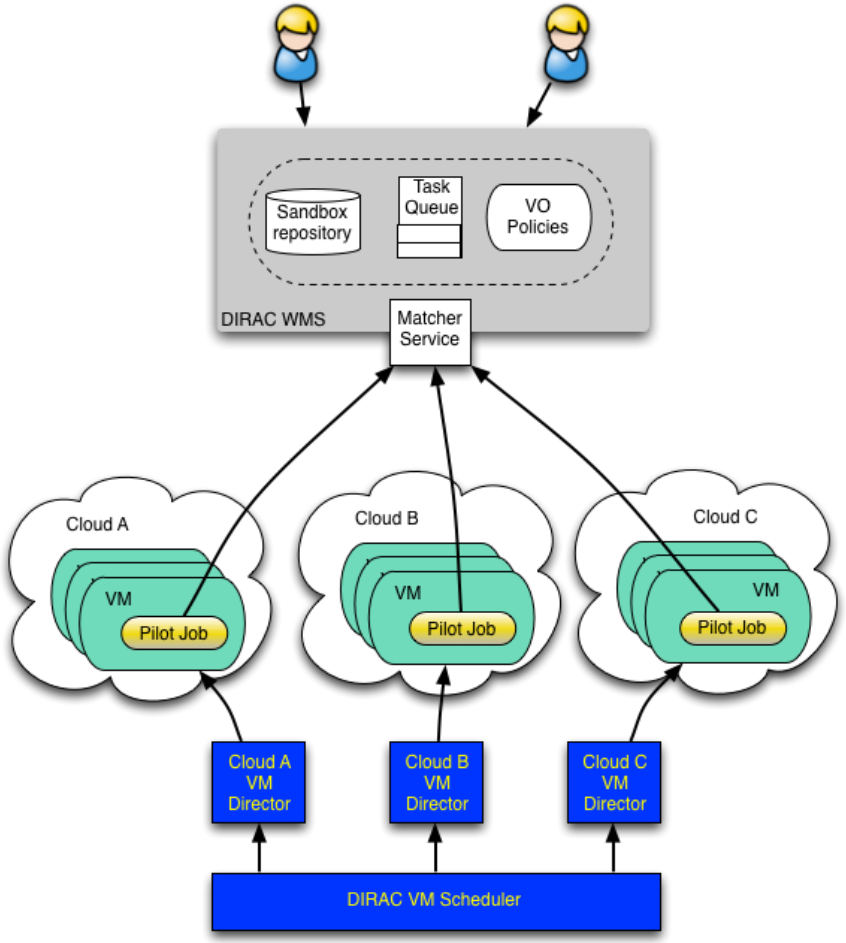
- **OpenNebula vs. OpenStack:**
  - Most preproduction and R&D activities going on use OpenStack as a Cloud Controller, Torino uses OpenNebula
  - Historical reasons: when we started OS was still unsuitable for production-level deployments
  - Fully satisfied with ONe, no reason or plans to switch at the moment
  - **Interoperability** is ensured by using **standard EC2 APIs** wherever possible
  - **"Biodiversity" is an asset!** (can bring ONe expertise to DataCloud, for example)

- **Administrative model:**
  - **Common procurement** is more efficient; purchases are driven by large WLCG Tier-2 tenders, others join in and can contribute even with very small amounts
  - It's easier to fund the infrastructure (growth and maintenance) if it is shared among several tenants
  - However, infrastructure and maintenance funding model is still hazy

VMDIRAC Spares

VMDIRAC: cloud integration with DIRAC

- "VM director" instead of "Pilot director"
  - start VMs, instead of submitting pilot jobs
- VMs at boot time start "pilot job"
  - This makes the instantiated VMs behave just as other WNs with respect to the DIRAC WMS
- VM scheduler need to manage dynamic virtual machines according to job situation



VMDIRAC: cloud integration with DIRAC

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

- 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
  - Automatically shutdown VMs when jobs stack is getting empty for a certain time

VMDIRAC: architecture and components

- Dirac server side
  - VM Scheduler – get job status from TQ and match it with the proper cloud site, submit requests of VMs to Director
  - VM Manager – take statistics of VM status and decide if need new VMs
  - VM Director – connect with cloud manager to start VMs
  - Image context manager – contextualize VMs to be WNs
- VM side
  - VM monitor Agent – periodically monitor the status of the VM and shutdown VMs when no need
  - Job Agent – just like “pilot jobs”, pulling jobs from task queue
- Configuration
  - Use to configure the cloud joined and the image
- Work together
  - **Start/Kill VMs**
  - **Run jobs on VMs**

B3CT Spares

B3CT: Kickstart.cfg

A kickstart file contains the information needed to perform the installation in order to **avoid the user providing them.**

Some parameters may be left to user input
(network parameters, keyboard layout, …).

It is possible to specify **packages or additional software** to be installed.

Software installed via kickstart:
OpenNebula, squid proxy, and rOCCI

B3CT: customized ISO

Starting from a **kickstart** and a **standard iso** it is possible to modify the iso so that it will look for the given kickstart at installation time.

Standard iso adopted for B3CT:

CentOS-6.7-x86_64-netinstall.iso

Care the **location of the kickstart**:

better to address the usb drive via its “label”.

B3CT: bootable Usb Key

Format (FAT32, mbr) a usb drive (2GB is enough).

Label must correspond to the **name** indicated in the customized iso creation process. Otherwise the kickstart will not be found when the installation begins.

Make the usb bootable:

- either via command line
- or using an application such as Unetbootin

B3CT: installation on Server

Start or reboot the machine and plug in the bootable usb drive.

From the BIOS menu choose the ***boot from usb drive*** option.

The installation will ask the user to provide the parameters not specified within the kickstart file, then the installation will proceed autonomously.

For instance, the user can provide:

- if the host will be a server providing services to the CI
- or will be one of the machines "only" hosting VMs for the worker nodes
- the network parameters

The machine will reboot when the installation is over.
Remove the usb drive and let the machine boot normally.

B3CT: test the Installation

Once the machine is up and running, users may control that the parameters have been properly set (network interface is up, storage is mounted, datastores interfaced …)

The entire installation process requires about 1 hour
(on the hardware used for the test).

The Sunstone hypervisor interface allows to verify that OpenNebula is working fine.

Validation proceeds creating virtual machines and:

- running jobs locally
- submitting jobs via BESIIIDIRAC

B3CT: virtual Machines

OS: SL5, SL6, Ubuntu 14

Test on local installation

Strict requirements on storage space

- the VM image has to be copied over the net each time a new VM is instantiated
- it could be a bottleneck depending on the requests

QCOW2 image format

- dynamic increase of the storage

Minimal OS installation

- only the required software is installed

B3CT: BESIII Software

Access to the BESIII software

- install while preparing the VM — huge space required (28GB)
- install during the contextualization — huge space (28GB) and time (2h)
- mount the sw via CVMFS — less space required (3.7GB)

The CVMFS downloads only the needed files

Software download time

- normally it has to be downloaded many times
- squid proxy allows to download the files once for better performances

B3CT: instantiating BESIII VMs

Copy the VM image to the server

Create a new image in OpenNebula

- a file .one will be provided
- the proper path have to be inserted

Create a new template in **OpenNebula**

- a file .txt will be provided
- the number of the image and the network have to be inserted

Instantiate a new VM

The machine is ready in less than 1 minute

The new VM can be used