VCondor - an implementation of dynamic virtual computing cluster

Yaodong Cheng, CC-IHEP, CAS

chyd@ihep.ac.cn

5-10 March 2017, Taipei (ISGC2017)

Problems

- Static resource management
  - Fixed CPU cores in each queue

- Resource access rights
  - Resources are owned by the group who invested
  - Only the group can use their own resources
  - Each person is allowed to submit some number of jobs. If one user submits too many jobs, these jobs will be queued

- Run environment of different experiments are not compatible with each other
  - Difficult to provide one physical machine to different experiments
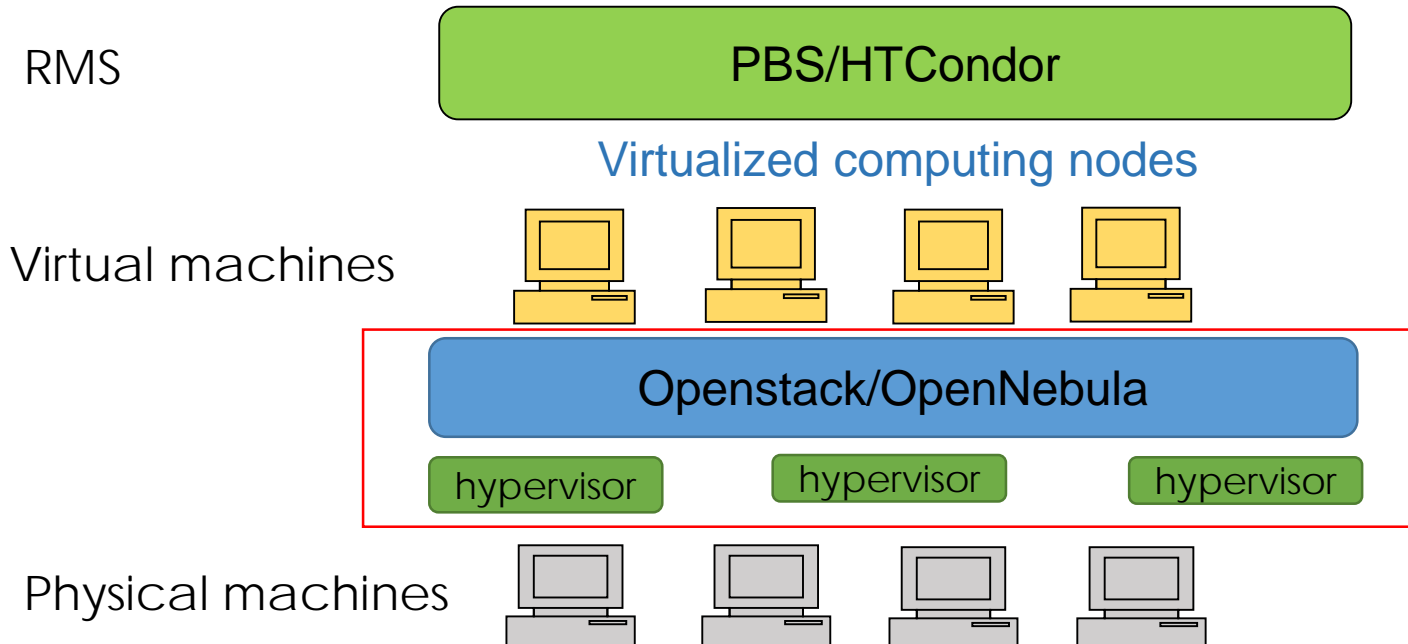
Virtualization and Cloud computing
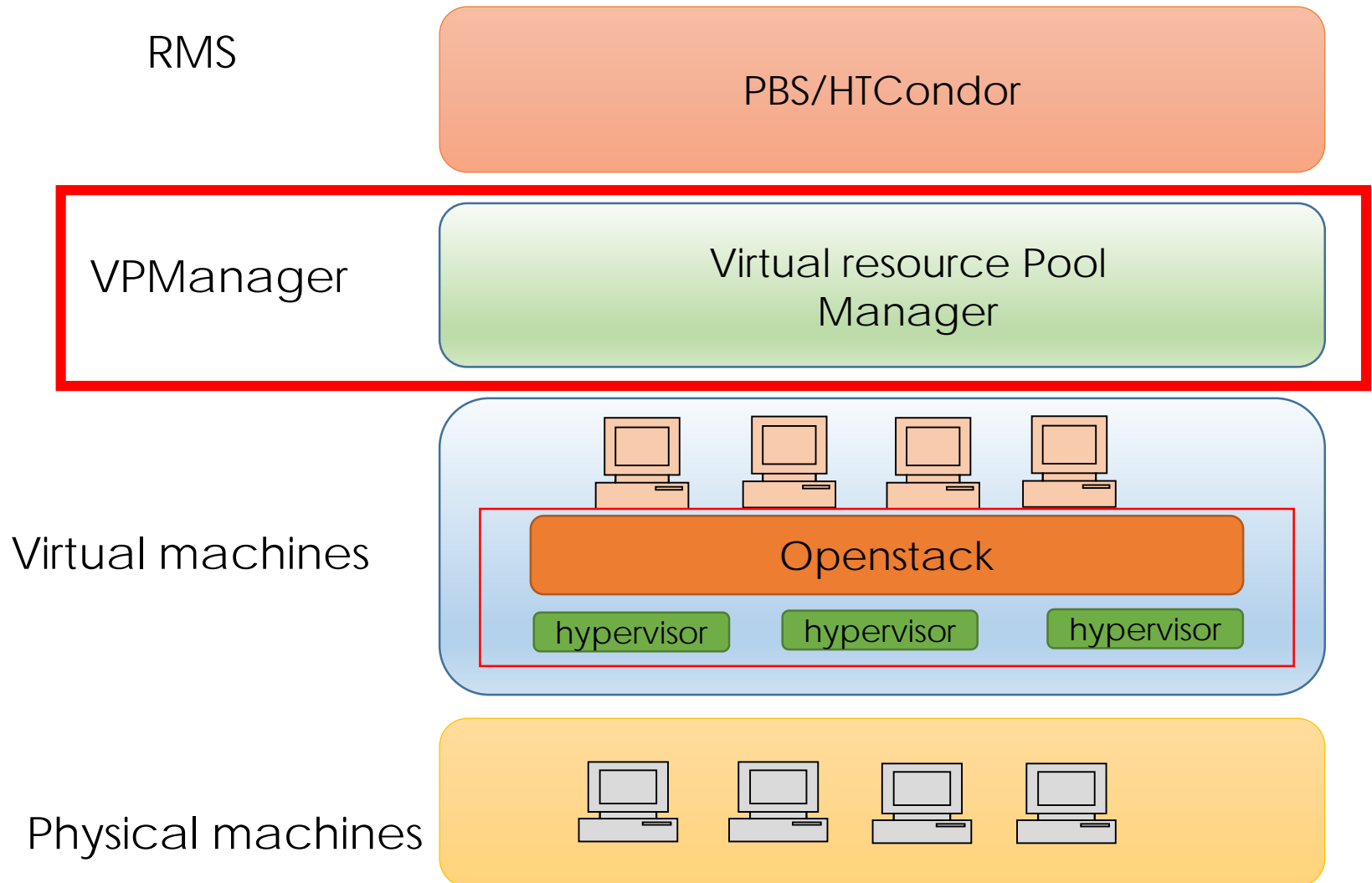
Motivation

GOALS we want to achieve

- Simplify the maintenance of computing resources
  - → Virtualization (virtual computing cluster)

- Share resources between different experiments
  - → Dynamic allocation of resources (dynamic virtual computing cluster)

- Meet the peak requirements of resources
  - → Integration of external resource transparent to user (cloud federation)
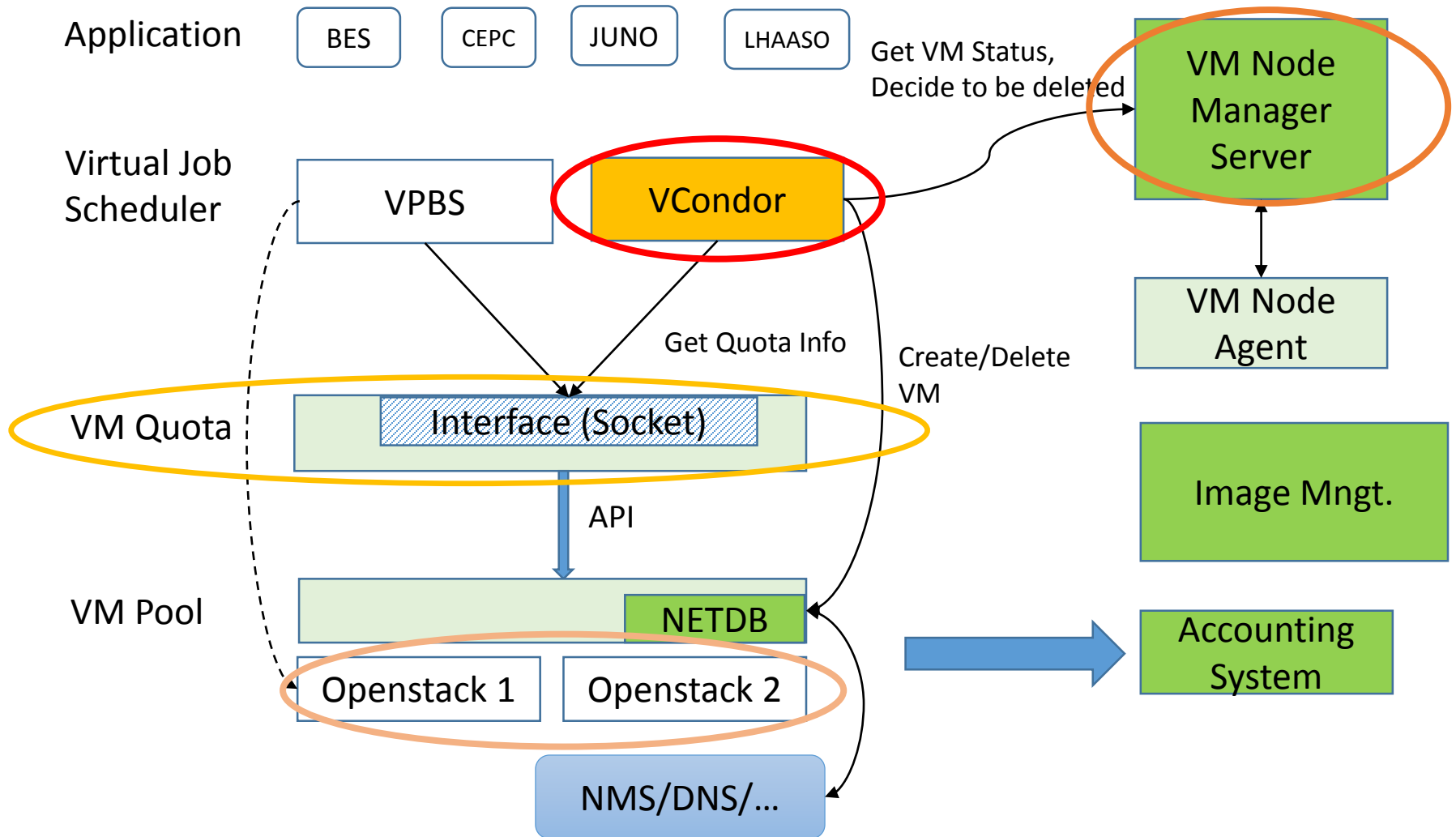
Virtual Computing Cluster

☐ Computing nodes are installed in virtual machines

☐ Seamless integration with the existing middleware stacks.

☐ Completely transparent to the computing service and end users

RMS | PBS/HTCondor

Virtualized computing nodes

Virtual machines

Openstack/OpenNebula

hypervisor   hypervisor   hypervisor

Physical machines

Dynamic virtual computing cluster

RMS — PBS/HTCondor

VPManager — Virtual resource Pool Manager

Virtual machines — Openstack / hypervisor / hypervisor / hypervisor

Physical machines

VPManager(Virtual resource Pool Manager)

Application: BES, CEPC, JUNO, LHAASO

Virtual Job Scheduler: VPBS, VCondor

VM Node Manager Server

Get VM Status, Decide to be deleted

VM Node Agent

Get Quota Info

Create/Delete VM

VM Quota: Interface (Socket)

API

Image Mngt.

VM Pool: NETDB

Openstack 1, Openstack 2

Accounting System

NMS/DNS/…

VPManager components

- VM Pool
    - manages one or more openstack instances, which hides the detailed information of openstack from upper applications
    - makes it possible to deploy multiple and different versions of openstack

- VM Quota
    - checks the information of VM Pool and requirements of different applications to allocate or reserve resources.

- Virtual job manager, VPBS and VCondor
    - checks the status of different queue and get the available VM number and create new VMs or destroy existing VMs.
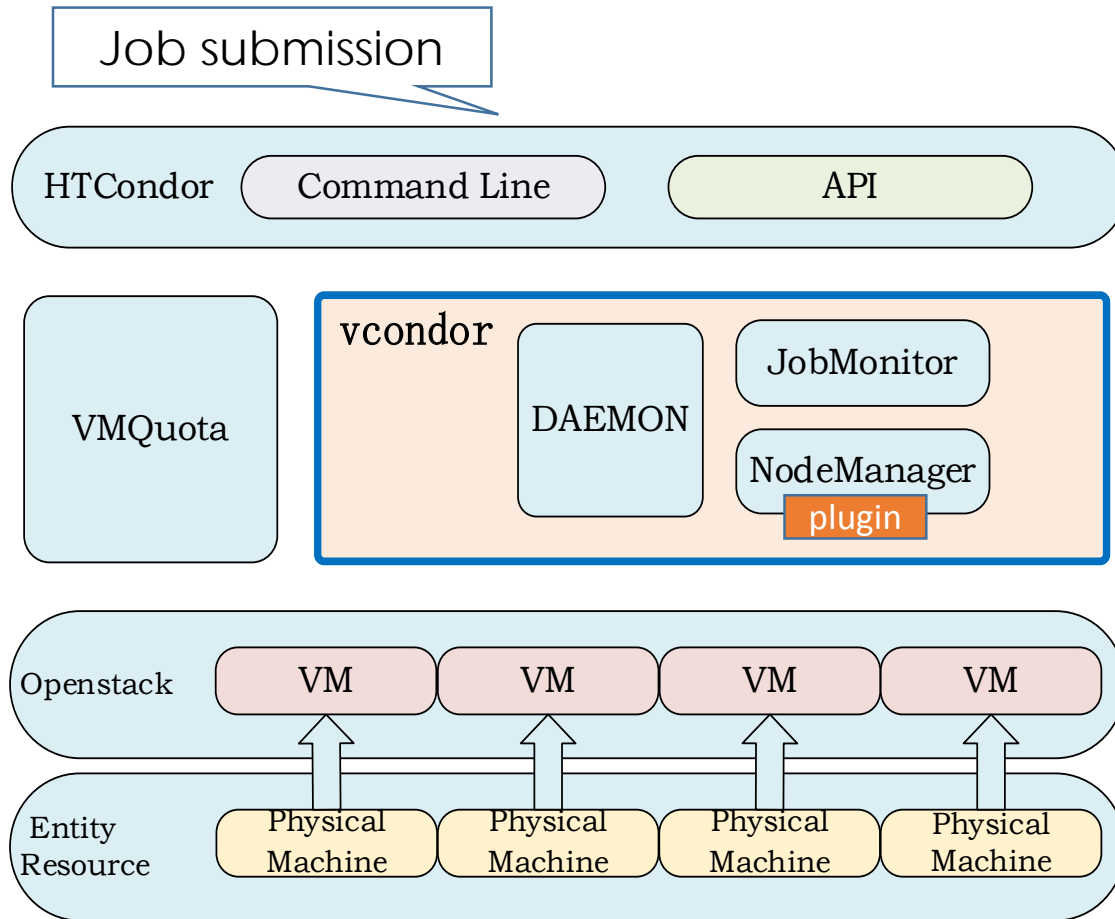
- VM node manager
    - checks and controls all the VM run environment such network status, affiliated job queue by an agent running in the virtual machine

- Accounting system
    - keeps all the usage information of each virtual machine and generate bills to user

VCondor

Job submission

| HTCondor | Command Line | API |

vcondor

DAEMON

JobMonitor

NodeManager
plugin

VMQuota

Openstack: VM VM VM VM

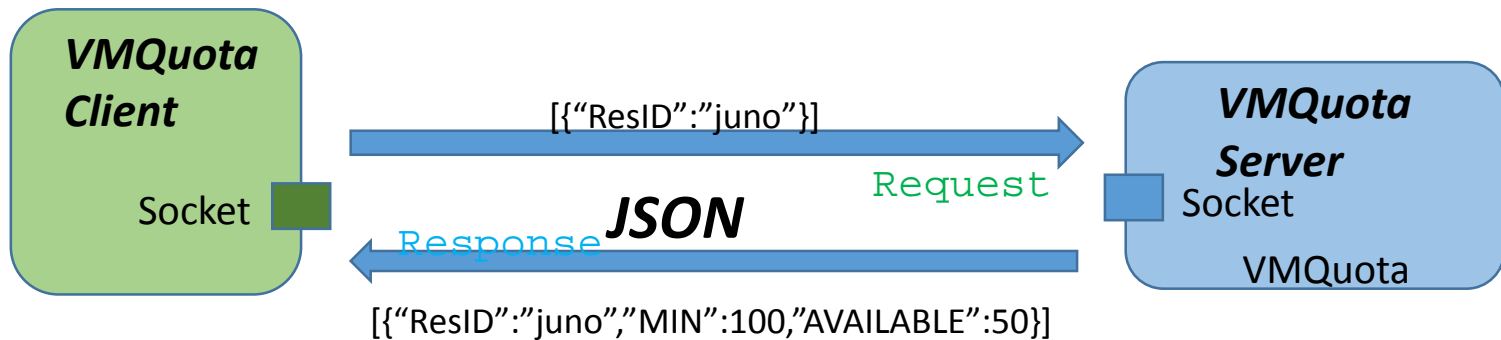Entity Resource: Physical Machine, Physical Machine, Physical Machine, Physical Machine

I. JobMonitor: query and record job information and HTCondor queue changes

II. NodeManager: use plugins (openstack api occi, or other interface) to create or destroy virtual machines
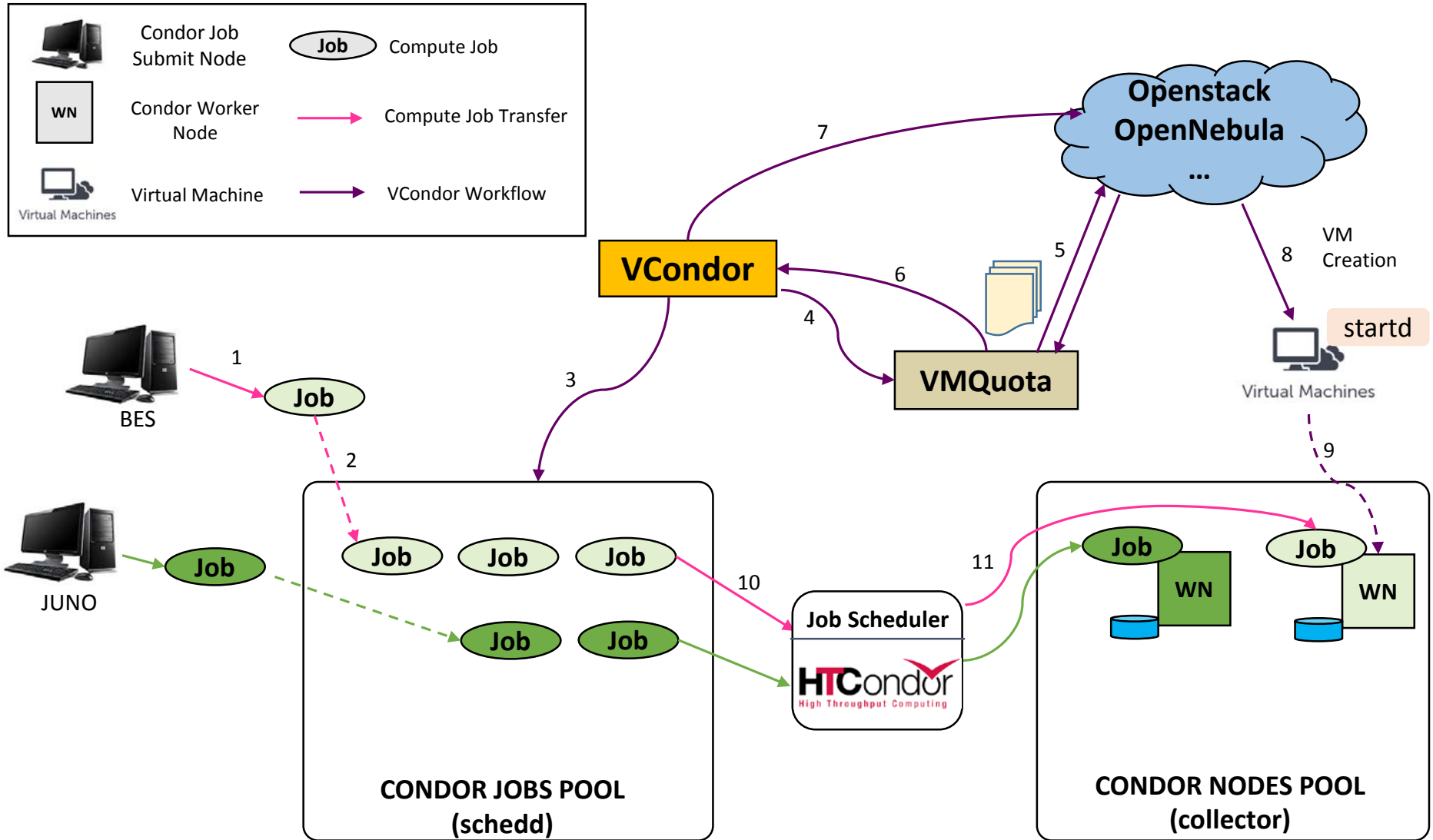
III. DAEMON: Main module, periodically executed

VMQuota

☐ Resource Quota management for different experiments

☐ Different experiments have different resource queues
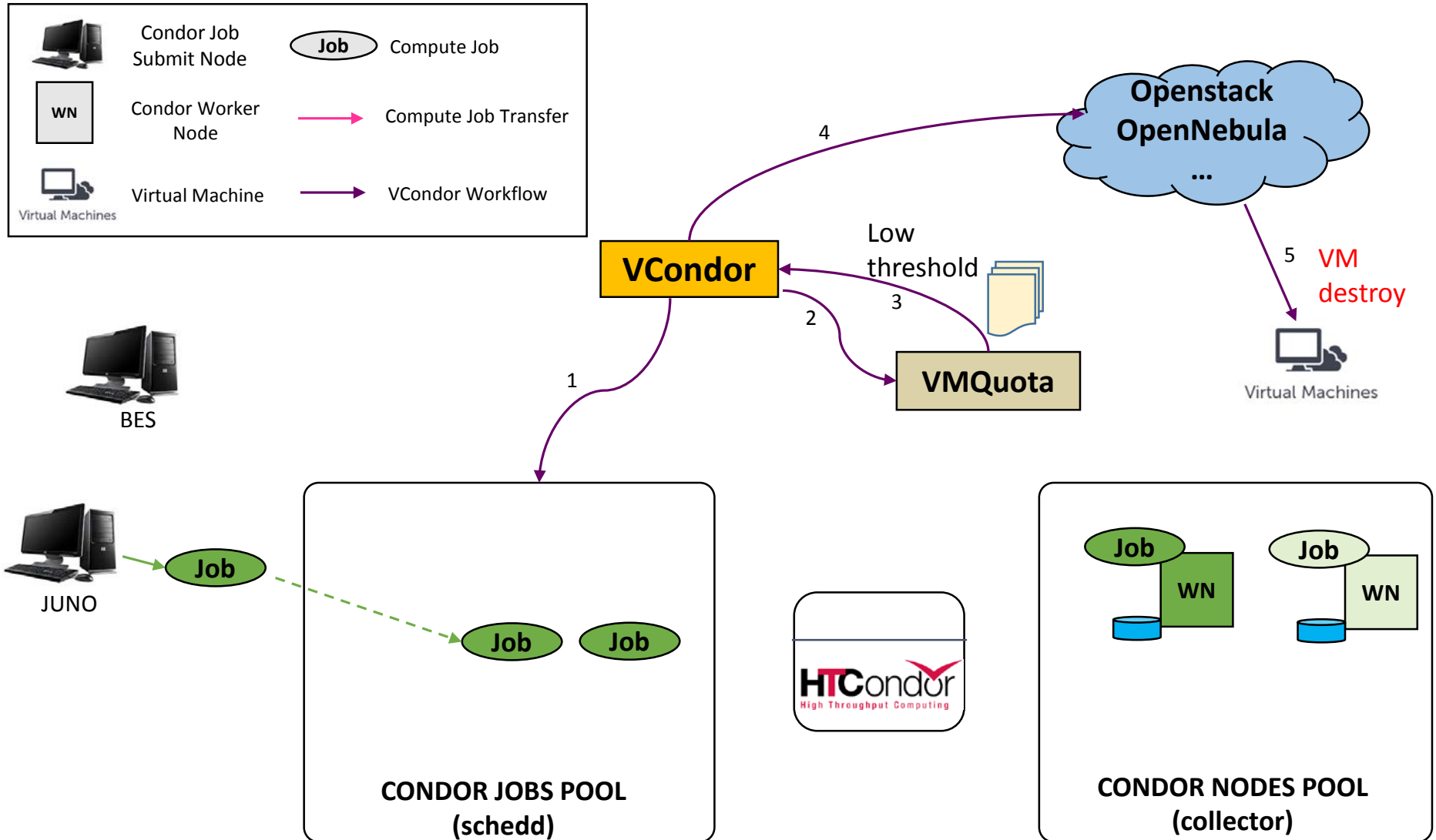
☐ Allocate and reserve resources for different queues

Queue Name	Low threshold	High threshold	Available	Reserve time(s)
BES	100	400	200	600
JUNO	100	300	150	600

VMQuota Client

Socket

[{"ResID":"juno"}]

Request

JSON

Response

[{"ResID":"juno","MIN":100,"AVAILABLE":50}]

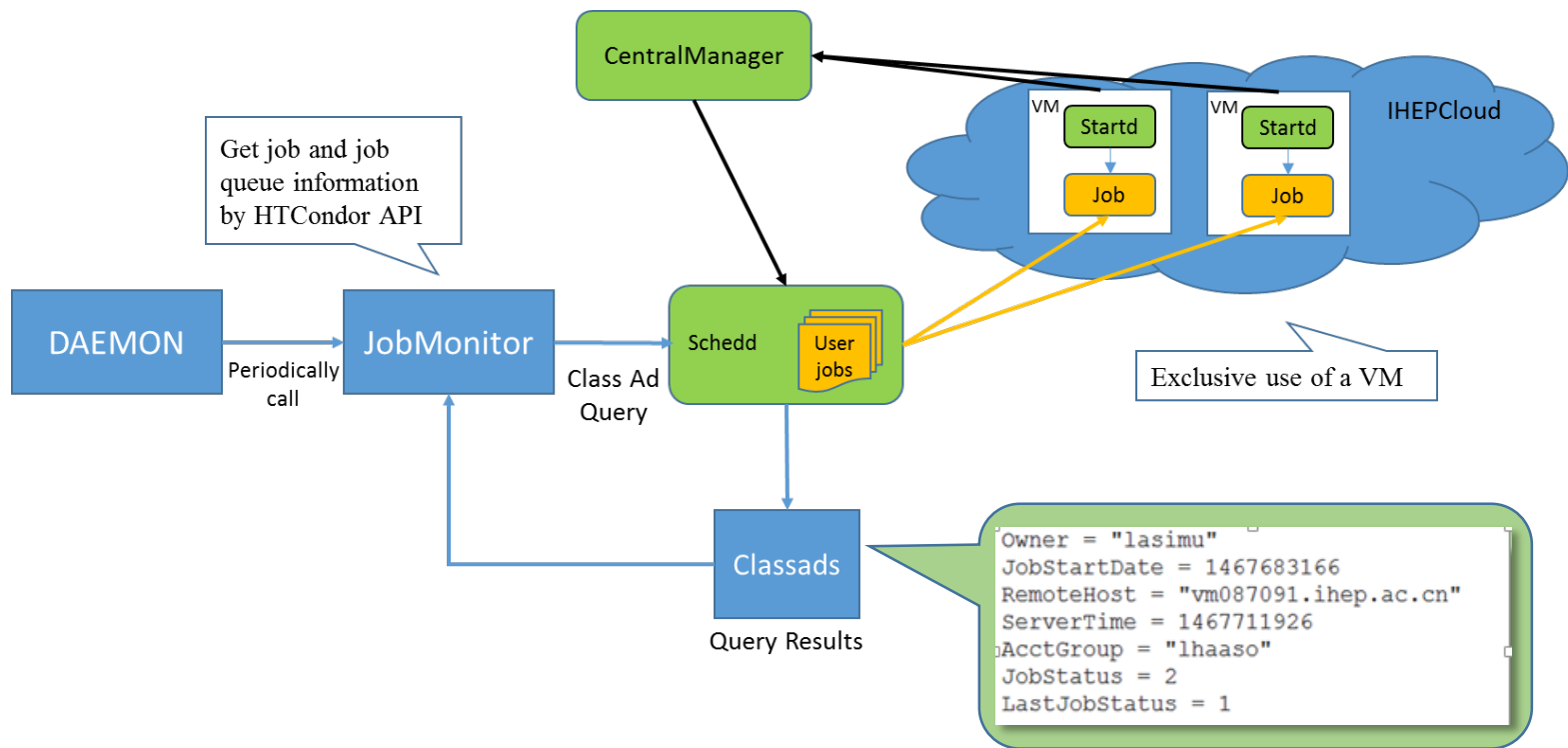**VMQuota Server**

Socket

VMQuota

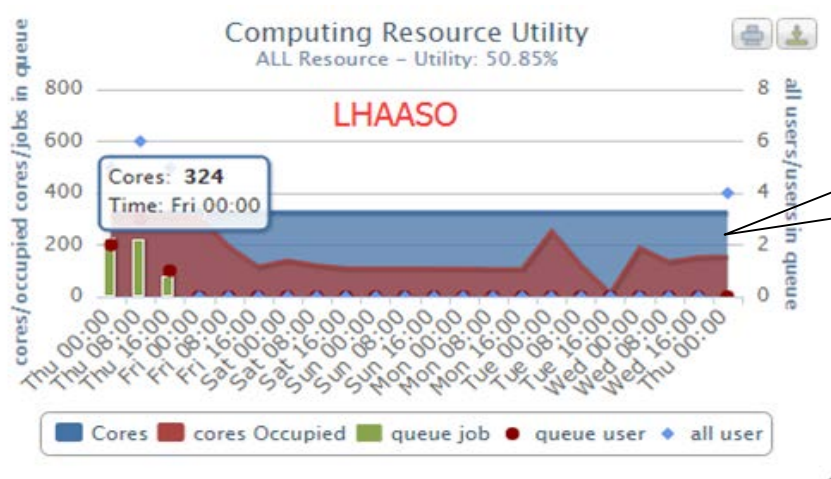# Resource pool expansion

Resource pool shrink

Deployment

@IHEP, China

- Virtual computing cluster, ~ 1000 CPU cores

- HTCondor

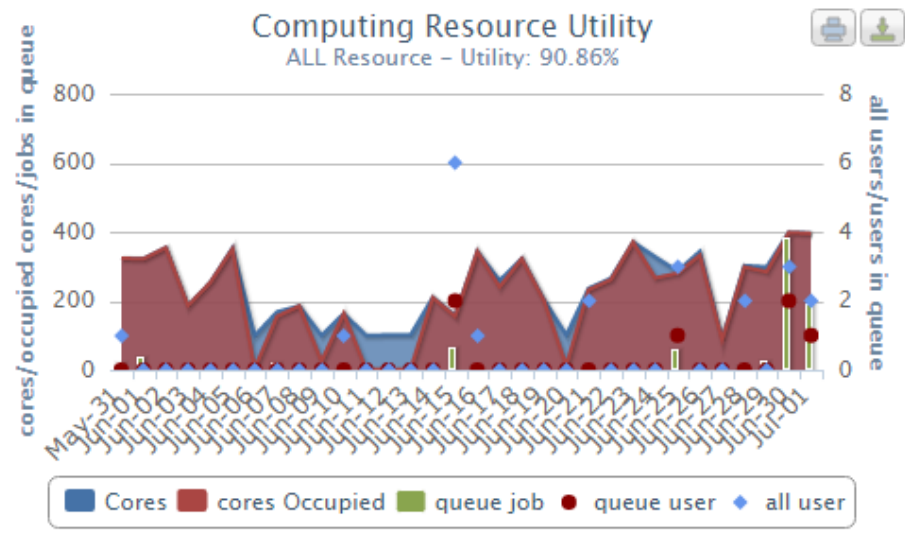- IHEPCloud: Openstack Kilo
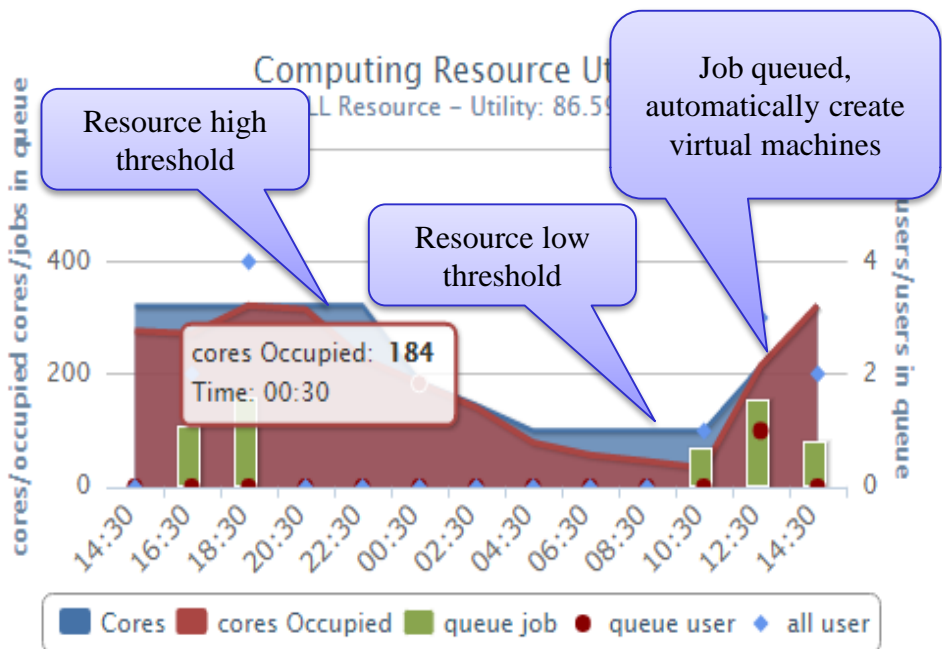
- Support LHAASO, JUNO, BES, CEPC, …

@JINR, Russia

- testbed

- HTCondor

- OpenNebula

VCondor monitoring

13

VCondor: How to use

☐ Download VCondor from https://github.com/hep-gnu/VCondor.git

GitHub

☐ Make sure HTCondor and Openstack or OpenNebula are well configured

☐ Setup a VM Image with HTCondor startd setup pre-configured

☐ Setup a VM Template with Image in the above

☐ The VCondor configuration file allows us to configure most of its functionalities

☐ Start VCondor and submit jobs, then resource pool scale up and down dynamically on-demand

Ongoing Work: cloud federation
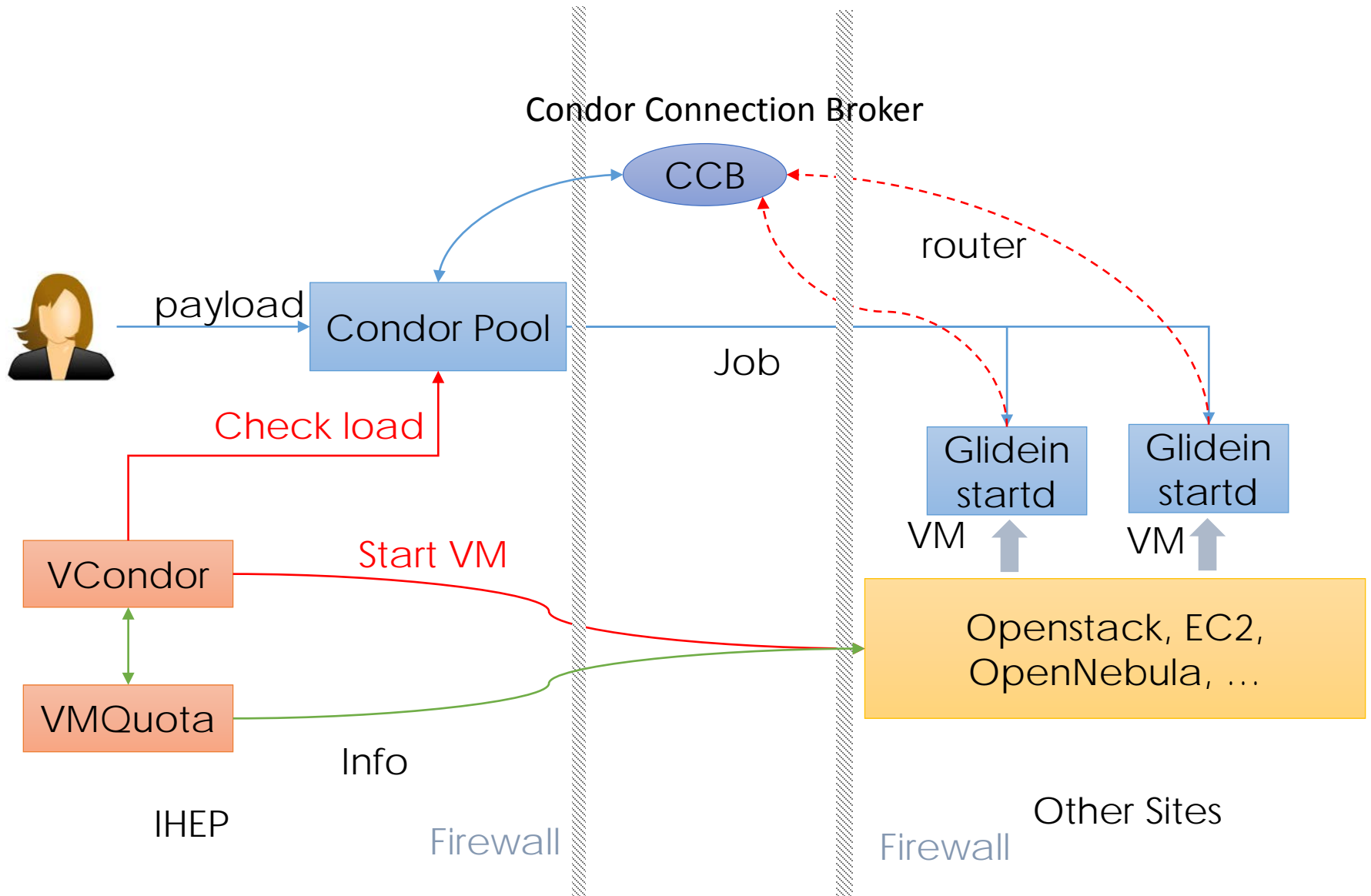
Conclusion

- Traditional cluster use static resource management, which leads to some problems
- VCondor is a middleware for dynamic virtual computing cluster
  - add or remove computing nodes depending on queue load
  - Improve resource utilization by sharing between different experiments
- VQuota coordinates the allocation of resources to make sure the fair and efficient use
- VCondor and HTCondor glidein is an optional solution to integrate remote cloud resources

Thank you!

Any Questions?