BESIII distributed computing and VMDIRAC

Xiaomei Zhang

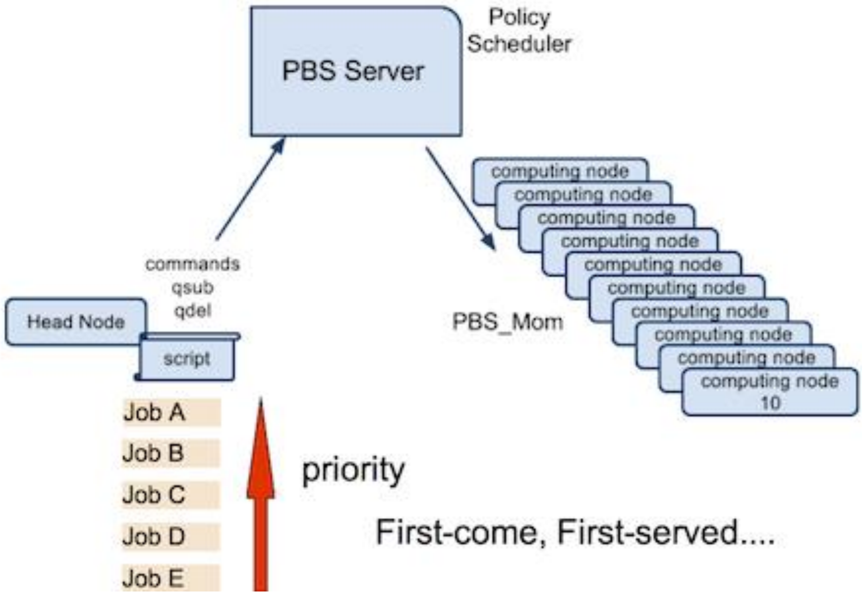
Institute of High Energy Physics
BESIII CGEM Cloud computing Summer School
Sep 7~ Sep 11, 2015

Content

- Two ways of scientific applications using cloud resources
  - VMDIRAC is an elastic way for the BESIII application to use cloud
- A real case： BESIII distributed computing
  - built up on DIRAC, VMDIRAC is a cloud extension
  - BESIII users use cloud through this platform
  - Demo : How to submit a job to Cluster and Grid, Cloud
- How VMDIRAC integrate cloud?
  - DIRAC workload management
  - VMDIRAC architecture and implementation

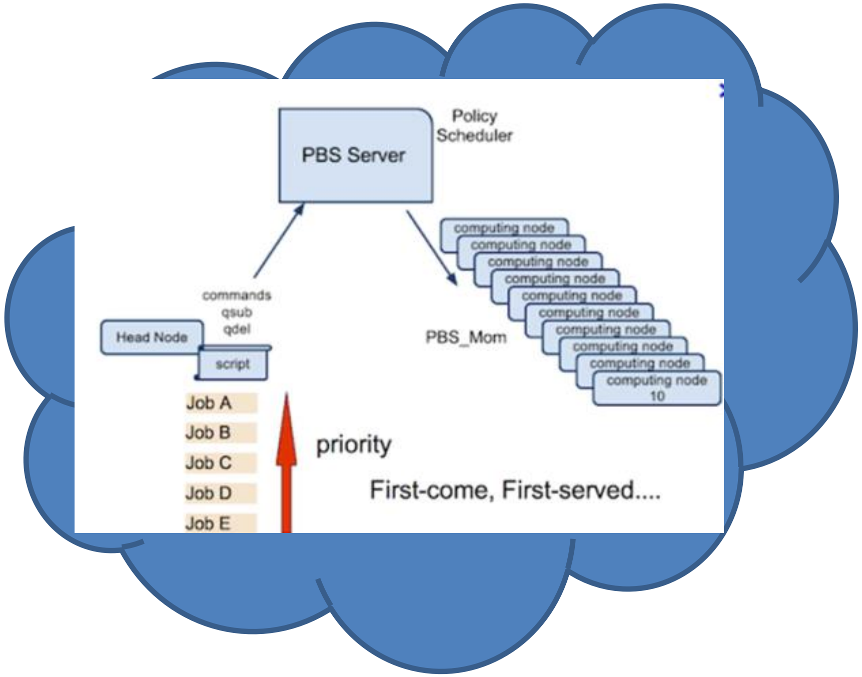
Run scientific applications on clusters

- The feature of Scientific applications
  - Enormous data processing with thousands of jobs to submit and run
- The most common way is to use resource manager to schedule these jobs to proper work nodes
  - PBS, HTCondor, LSF.....



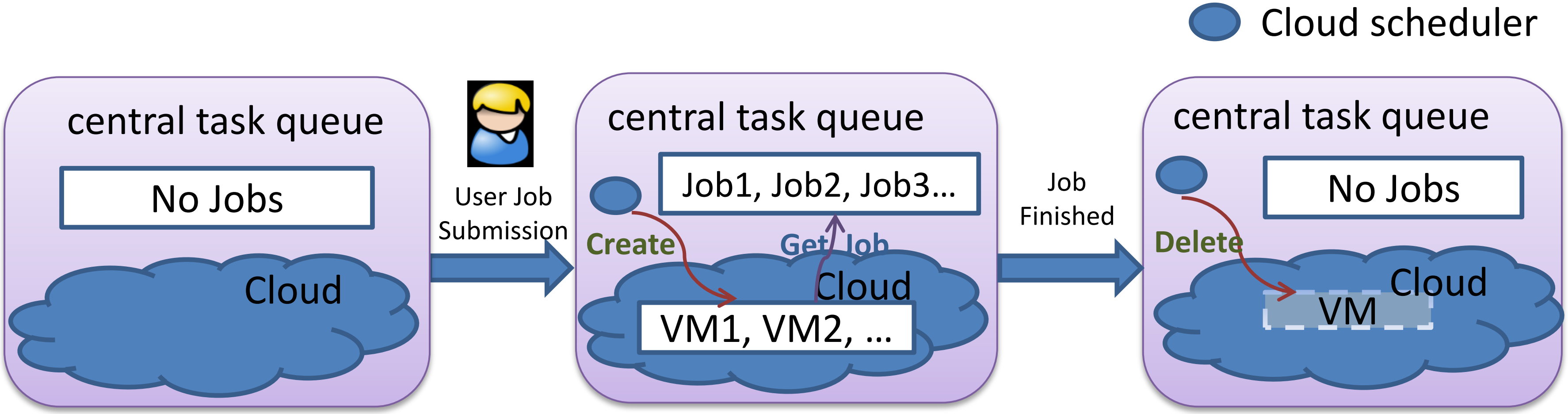
Run scientific applications on clouds

- Build standalone virtual cluster over cloud
  - Everything built over VMs instead of physical machines
  - Transparent to end users
  - Easier, not so flexible
- Based on contextualization technique, we can automatically set up a virtual cluster with “one button”
  - “cernvm-online” in yesterday stefano’s talk and demo



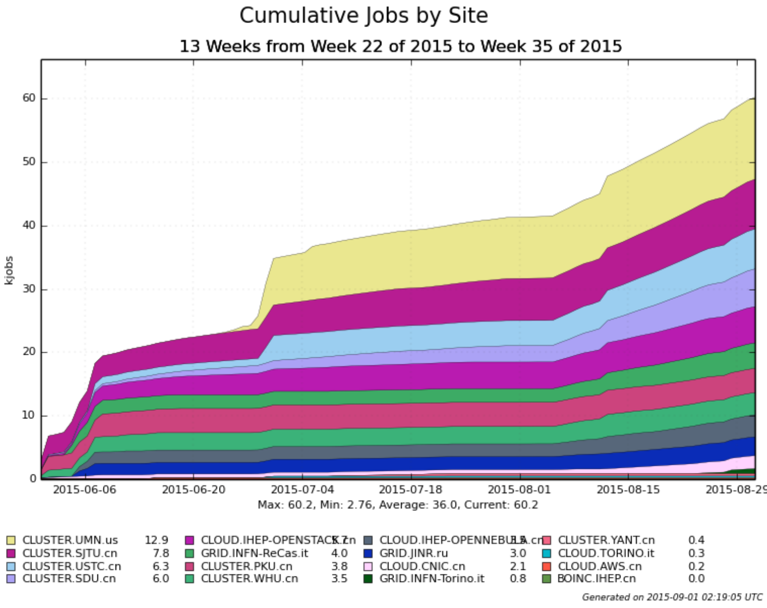
Run scientific applications on clouds

- 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



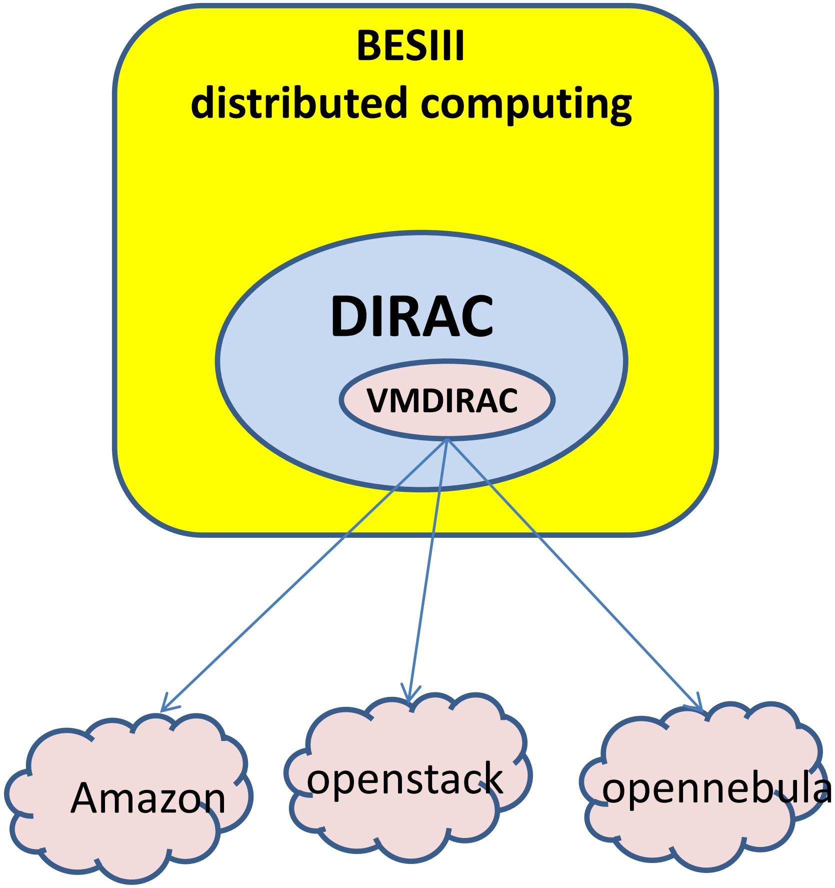
BESIII distributed computing

- BESIII distributed computing system provides a way for BESIII physics users to use various distributed computing resource
  - Grid, Cluster, Cloud and Volunteer computing
  - more than 14 sites are joined
  - About 2000 cores CPU resources, 400 TB storage have been integrated
- 60K jobs have been submitted and run over distributed computing resources in recent three months



BESIII distributed computing

- Use CVMFS to deploy BESIII experiment software to remote sites
- The system is built up based on DIRAC
- VMDIRAC is a cloud extension of DIRAC
  - Able to integrate both private cloud and commercial cloud, eg. openstack, cloudstack, opennebula, etc



Authentication on BESIII distributed computing

- As a BESIII user, you are allowed to submit jobs to resources
- DIRAC use grid certificate to check if you belong to BESIII
  - First you need to get certificate from one of grid CA (Certification Authority)
    - IHEP CA is the only one in China (https://cagrid.ihep.ac.cn)
  - Second you have to register your certificate in BESIII VO(Virtual Organization)
    - https://voms.ihep.ac.cn

```
-bash-4.1$ voms-proxy-info -all
……
=== VO bes extension information ===
VO        : bes
subject   :
/C=CN/O=HEP/OU=CC/O=IHEP/CN=Xiao
mei Zhang
issuer    :
/C=CN/O=HEP/OU=CC/O=IHEP/CN=vom
s.ihep.ac.cn
attribute :
/bes/Role=NULL/Capability=NULL
timeleft  : 11:59:46
uri       : voms.ihep.ac.cn:15001
```

Demo: How to submit jobs through DIRAC web portal

- Check the permission to use the resources
  - https://dirac.ihep.ac.cn
- Check the available resources
  - https://dirac.ihep.ac.cn:8444/DIRAC/CAS_Production/user/jobs/SiteSummary/display
- Submit a job to resources including cloud
- Monitor job running status
- Get the results from jobs

How to submit jobs to cloud through DIRAC client

- More complicated applications can use command line to submit jobs
  - Source DIRAC environment
  - Initialize your grid certificate to get permission
  - Prepare JDL files
  - dirac-wms-job-submit *.jdl
  - dirac-wms-job-get-output <jobID>

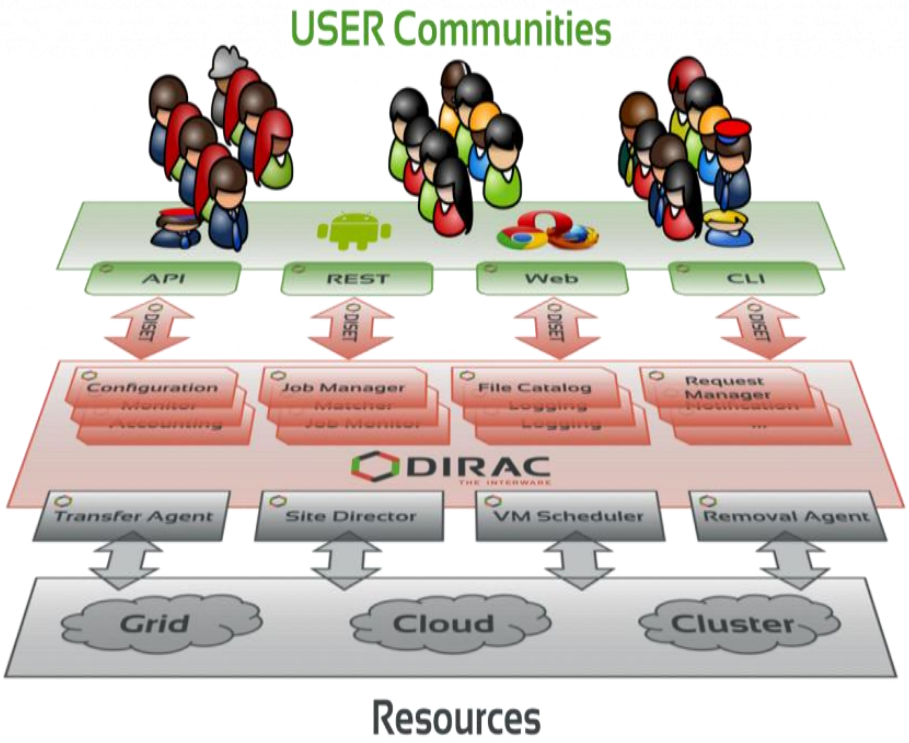
```
[
 Executable = “/bin/ls";
 JobRequirements =
 [
   CPUTime = 86400;
   Sites = "CLOUD.CNIC.cn";
 ];
 StdOutput = "std.out";
 StdError = "std.err";
 OutputSandbox =
 {
   "std.err",
   "std.out"
 };
]
```

DIRAC

- Distributed Infrastructure with Remote Agent Control
- History
  - DIRAC project was born as the LHCb distributed computing project
  - Since 2010 DIRAC became an independent project
- DIRAC has all the necessary components to build ad-hoc infrastructures for distributed computing as a ***framework***
  - Configuration, agents, services, user interface, databases
  - Allow to customize experiment-specific systems

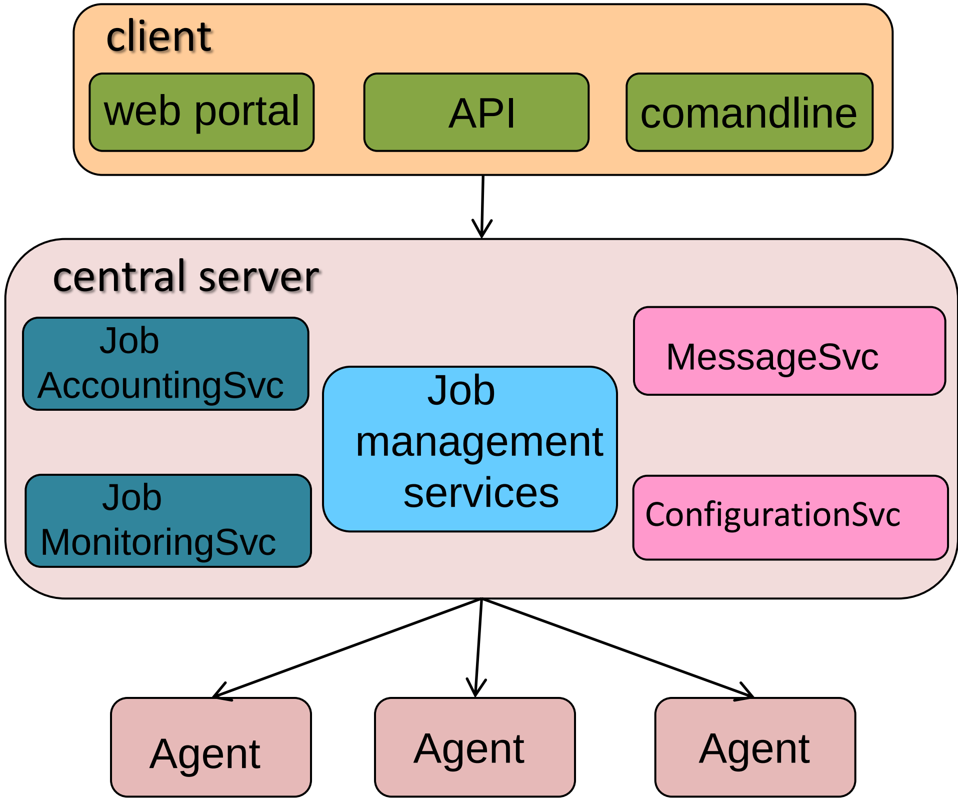
DIRAC

- 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, agents, clients, databases

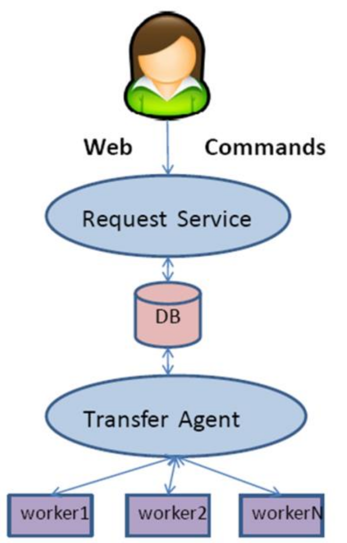


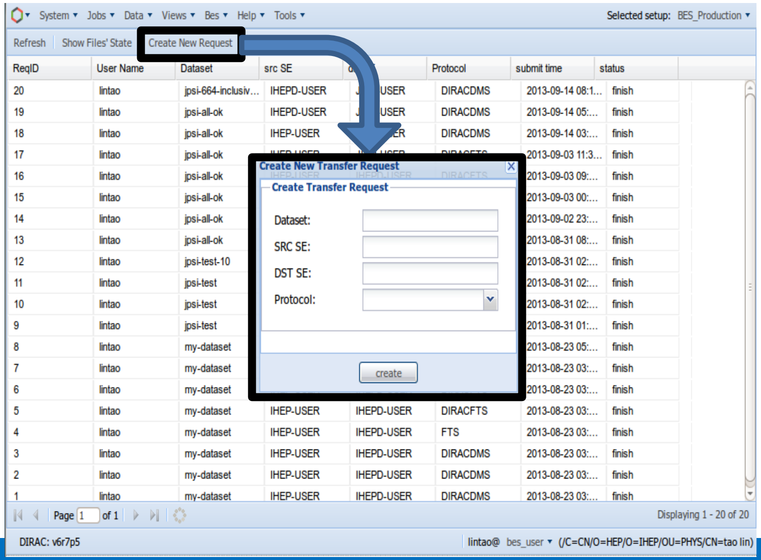
DIRAC systems

- 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

A case --- BESIII Transfer system

- Do mass transfers between remote sites
- The Components include:
  - Web interface
    - Request transfers
    - Monitor transfer status
  - Transfer agent
    - Get transfer tasks from DB
    - Start transfers
  - Request service
    - Get requests from users
  - DB
    - Record transfer requests and status
- VMDIRAC is another system in DIRAC, just more complicated





Selected setup: BES_Production

ReqID	User Name	Dataset	src SE	d	Protocol	submit time	status
20	lintao	jpsi-664-inclusiv...	IHEPD-USER		DIRACDMS	2013-09-14 08:1...	finish
19	lintao	jpsi-all-ok	IHEPD-USER		DIRACDMS	2013-09-14 05:...	finish
18	lintao	jpsi-all-ok	IHEP-USER		DIRACDMS	2013-09-14 03:...	finish
17	lintao	jpsi-all-ok				2013-09-03 11:3...	finish
16	lintao	jpsi-all-ok				2013-09-03 09:...	finish
15	lintao	jpsi-all-ok				2013-09-03 00:...	finish
14	lintao	jpsi-all-ok				2013-09-02 23:...	finish
13	lintao	jpsi-all-ok				2013-08-31 08:...	finish
12	lintao	jpsi-test-10				2013-08-31 02:...	finish
11	lintao	jpsi-test				2013-08-31 02:...	finish
10	lintao	jpsi-test				2013-08-31 02:...	finish
9	lintao	jpsi-test				2013-08-31 01:...	finish
8	lintao	my-dataset				2013-08-23 05:...	finish
7	lintao	my-dataset				2013-08-23 03:...	finish
6	lintao	my-dataset				2013-08-23 03:...	finish
5	lintao	my-dataset	IHEP-USER	IHEPD-USER	DIRACFTS	2013-08-23 03:...	finish
4	lintao	my-dataset	IHEP-USER	IHEPD-USER	FTS	2013-08-23 03:...	finish
3	lintao	my-dataset	IHEP-USER	IHEPD-USER	DIRACDMS	2013-08-23 03:...	finish
2	lintao	my-dataset	IHEP-USER	IHEPD-USER	DIRACDMS	2013-08-23 03:...	finish
1	lintao	my-dataset	IHEP-USER	IHEPD-USER	DIRACDMS	2013-08-23 03:...	finish

Page 1 of 1

Displaying 1 - 20 of 20

DIRAC: v6r7p5

lintao@ bes_user ▾ (/C=CN/O=HEP/O=IHEP/OU=PHYS/CN=tao lin)

Create New Transfer Request

Create Transfer Request

Dataset:

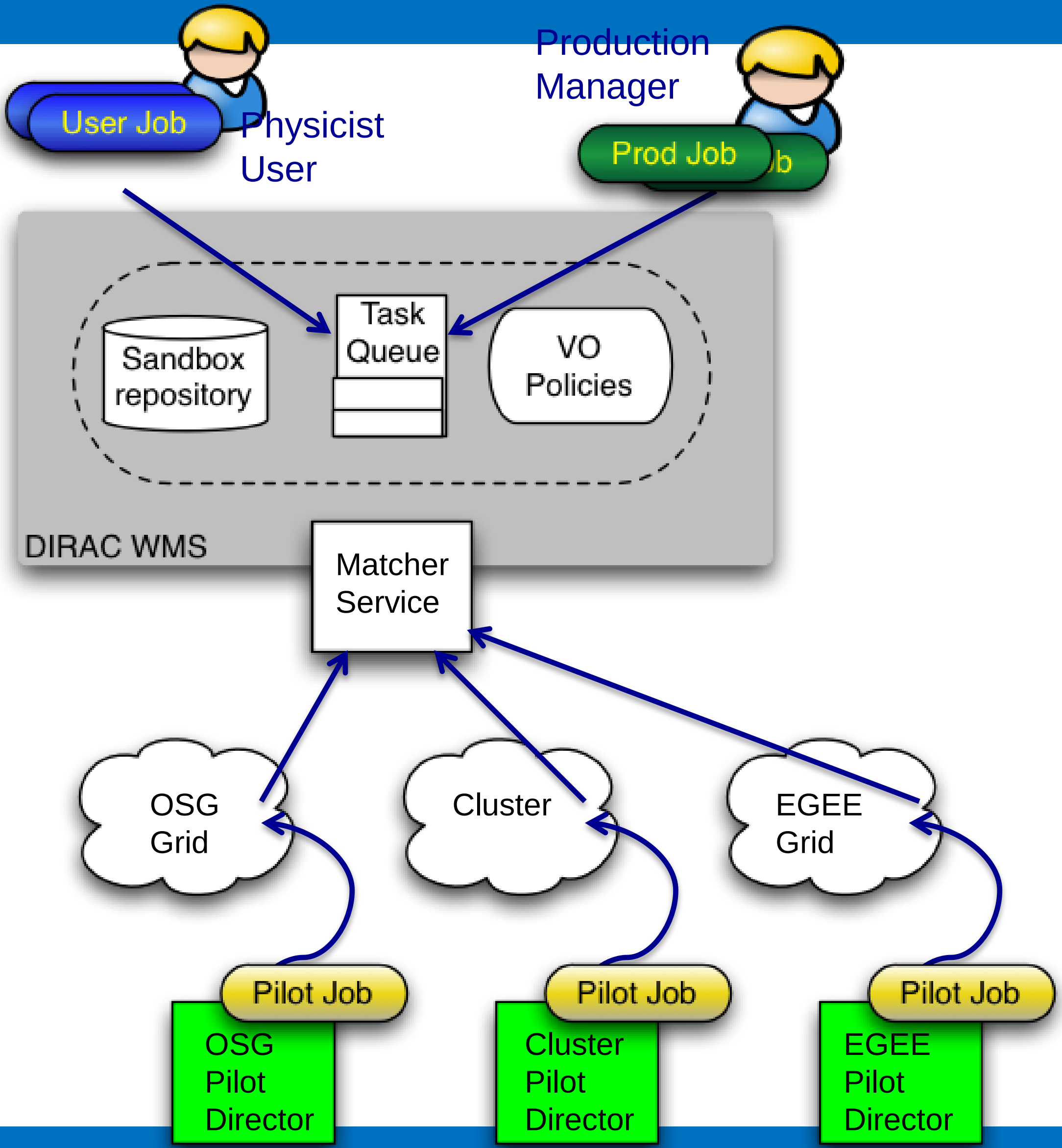
SRC SE:

DST SE:

Protocol:

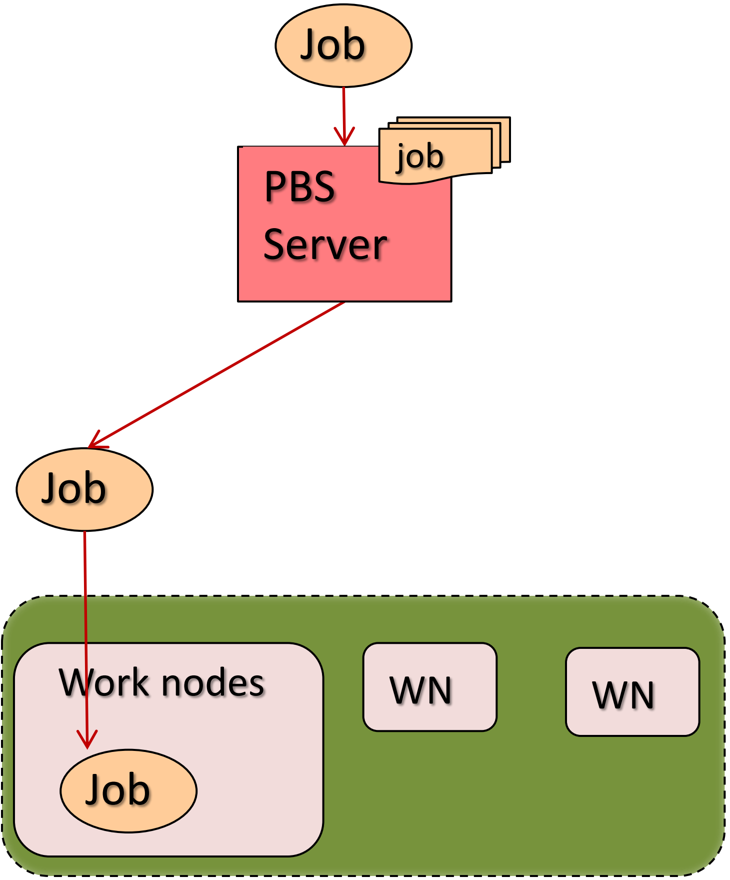
DIRAC workload management

- 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



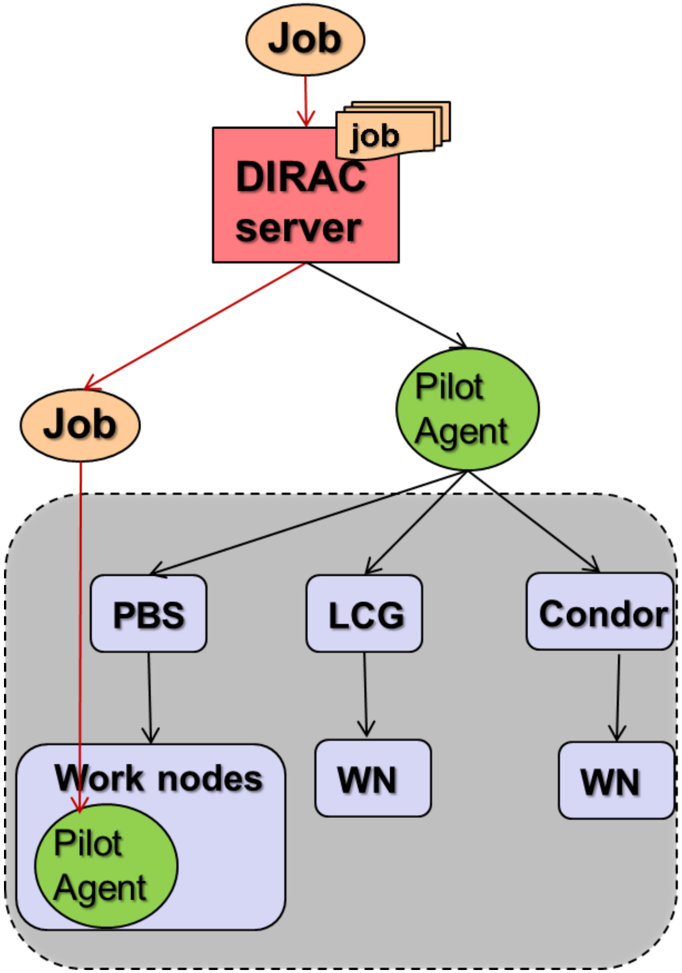
Push scheduling

- 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



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

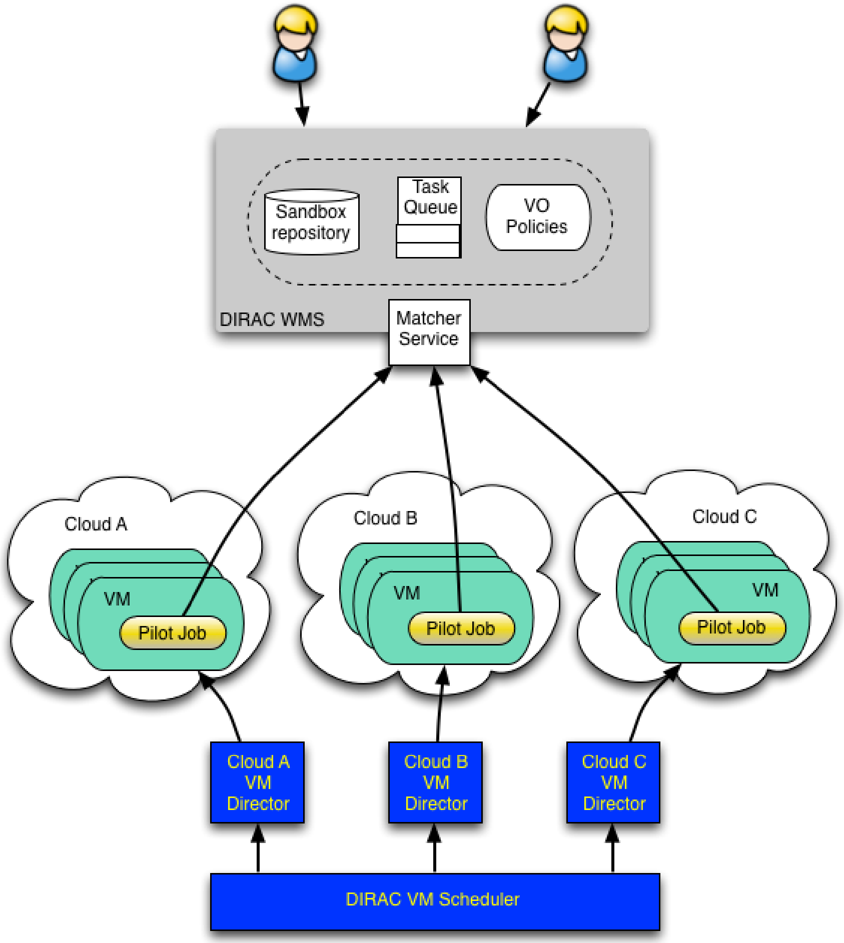


Cloud differences

- Cloud is integrated into DIRAC in similar way, but with some differences
- Local job scheduler and resource manager
  - Cluster: pbs, condor
  - Grid: arcCE, creamCE
  - Cloud: no, only cloud manager to control VMs
- Static and dynamic resources
  - Static WNs in Cluster and Grid
  - No WNs before jobs are coming

Cloud integration

- “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

- 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 no jobs need

Architecture and components

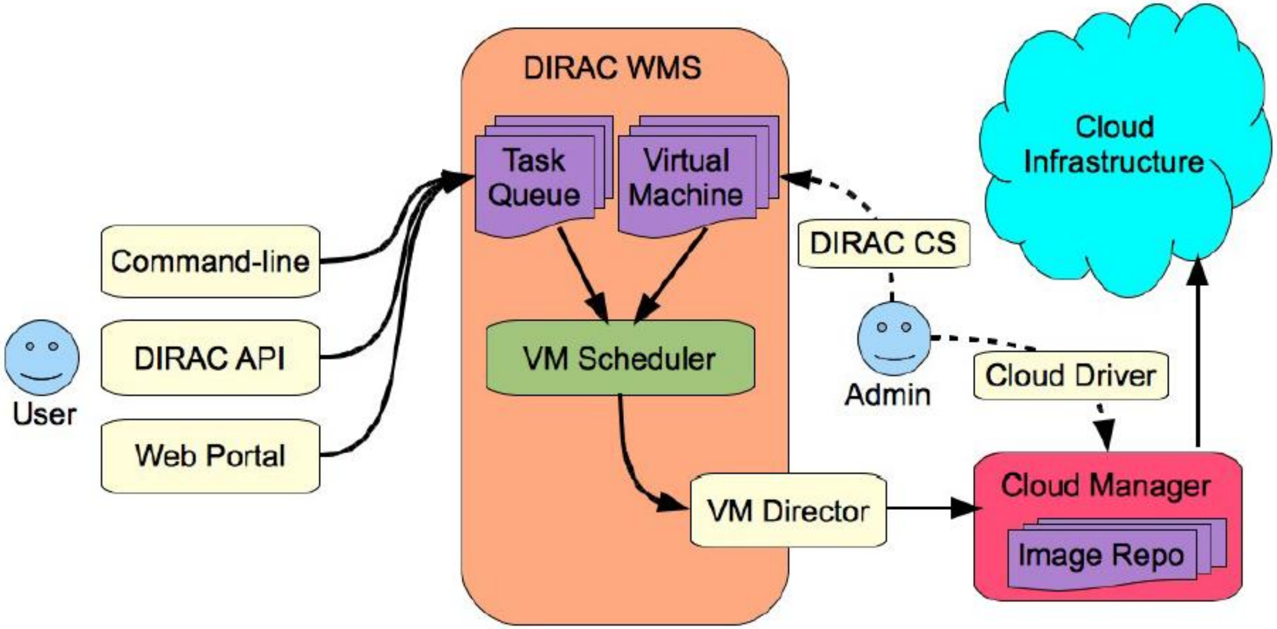
- Dirac server side
  - VM Scheduler – get job status from TQ and match it with the proper cloud site, submit requests of VMs to Director
  - VM Manager – take statistics of VM status and decide if need new VMs
  - VM Director – connect with cloud manager to start VMs
  - Image context manager – contextualize VMs to be WNs

Architecture and components

- 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 VMs
  - Run jobs on VMs

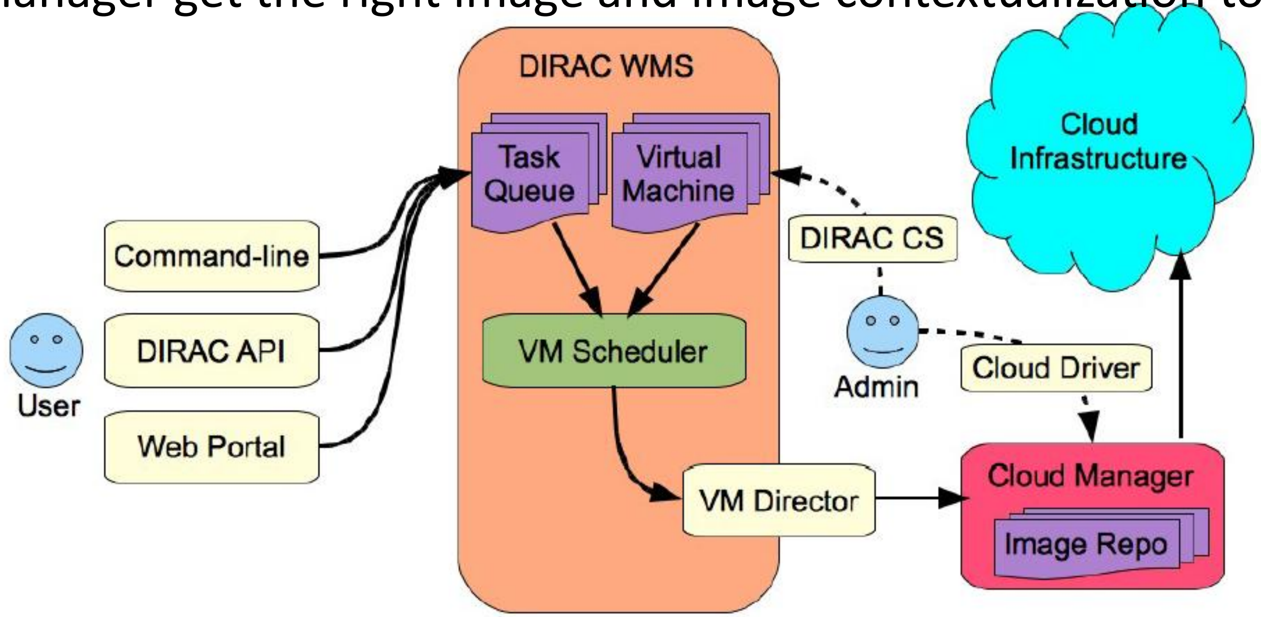
How to start VMs

- Users submit jobs through DIRAC interface
- Jobs recorded in task queue
- Cloud and VMs status recorded in the database
  - Cloud and images info get from DIRAC CS
  - DIRAC admin has uploaded the proper images in advance by cloud driver
  - VMs status is collected by VM managers



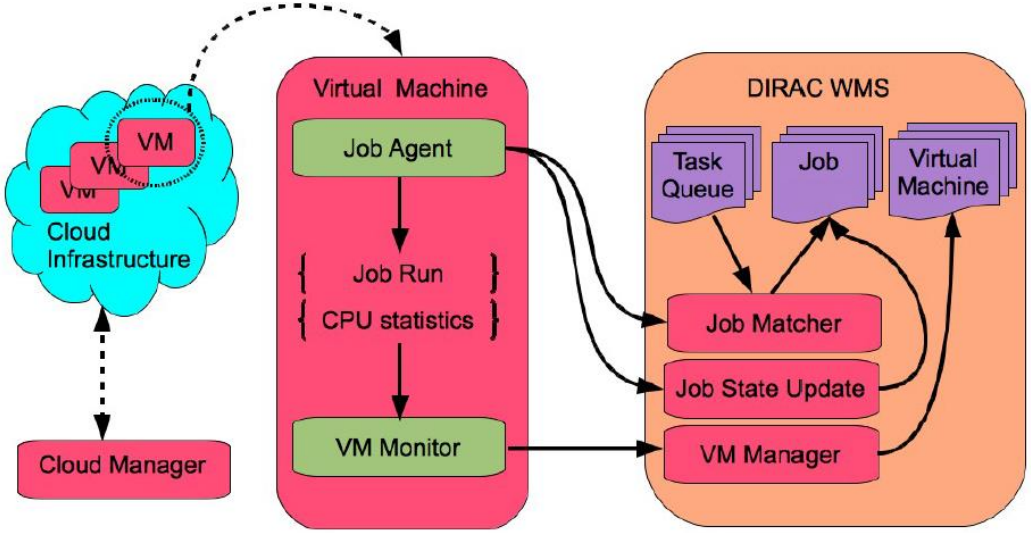
How to start VMs

- VM scheduler gets the list of jobs from the central Task Queues to run by matching the pending tasks with the available cloud
- VM scheduler also check if the existing VMs is enough with job info. If not enough and the maximum VMs threshold is not reached, then it submit a request of new VMs
- The proper VM director connect with Cloud Manager through Cloud API such rocci, libcloud, EC2…..
- Cloud manager get the right image and image contextualization to start VMs



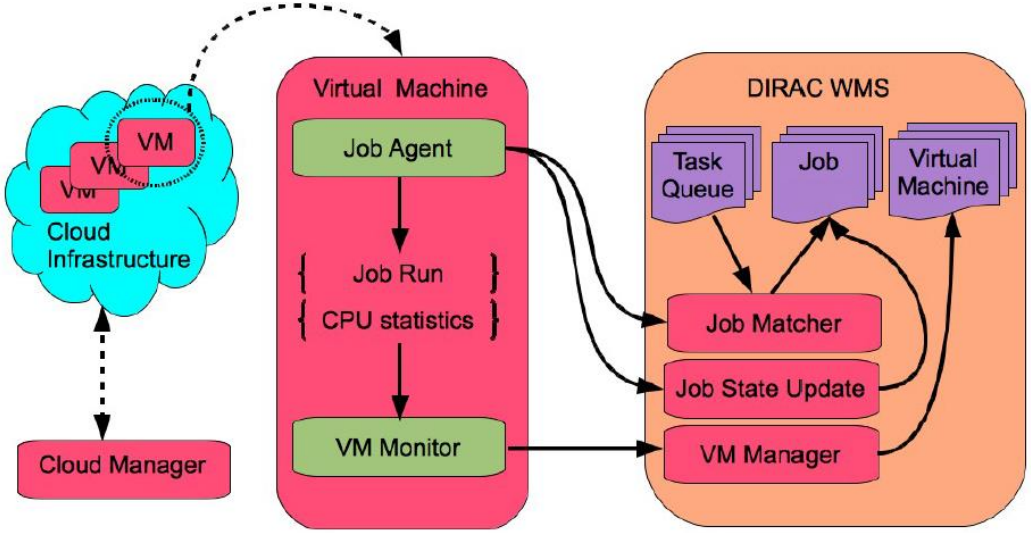
How VMs run jobs

- The VM started is a “full” VM
  - At boot time, it is contextualized and starts DIRAC job Agent and VM Monitor Agent
- Job Agent
  - Cooperate with Job Matcher, and get proper jobs from task queue
  - Start the jobs and supervise their correct execution on the Virtual Machine resource
  - Report periodically to Job state update agent to update job status in DB



How VMs run jobs

- VM monitor agent
  - Report VM running state to VM manager
  - Monitor the CPU load of VM, and when the load is dropped a certain threshold, the VM manager will halt VMs
  - The VM monitor also will help asynchronously uploads the output data when the VM takes new execution

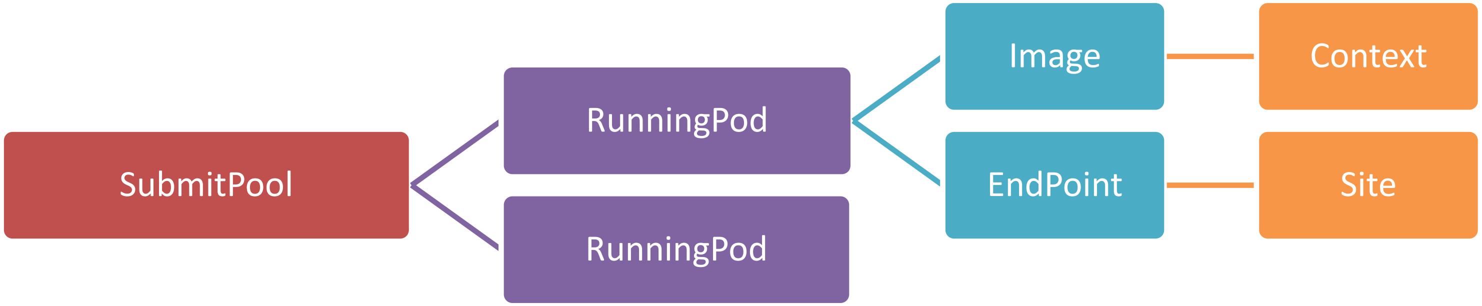


The contextualization mechanism

- The contextualization mechanism allows to configure the VM to start the pilot script at boot time
  - Avoid building and registering enormous number of images
- Ad-hoc image (no contextualization)
  - Install VMDIRAC staffs and security certificate in the images
  - Upload images to every cloud
- Contextualization supported for different cloud manager
  - Generic SSH
  - HEPIX OpenNebula
  - Cloudinit

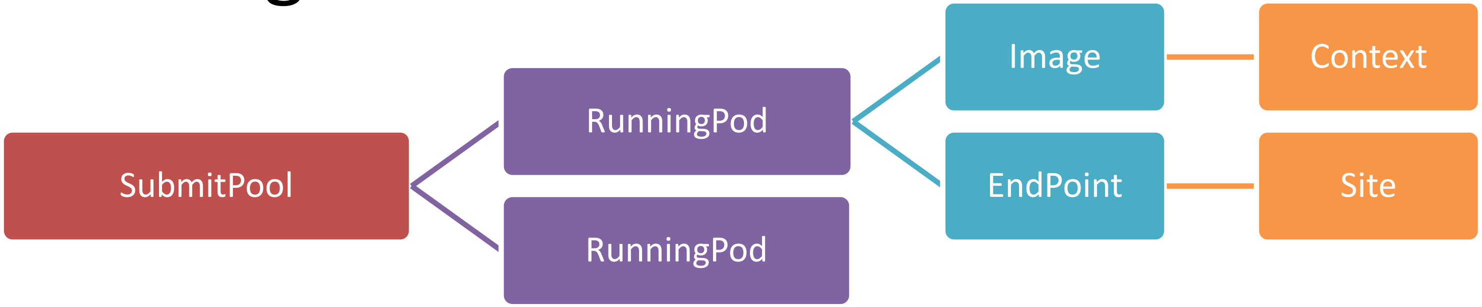
VMDIRAC configuration

- Collect info of the available clouds and images
- “Endpoint” is used to define the cloud endpoint
- “Image” is to tell you the running env the VM is going to provide
  - Here “image” includes the selection of contextualization methods



VMDIRAC configuration

- “Running Pods” match “Endpoint” and “Image” to define various running conditions
  - Every cloud properly need the special image and contextualization methods
    - Security reason, special format, etc
- “Submit pools” is to collect the info of “Running Pods” for VM Scheduler to choose



VM monitor

- **Central monitor**
  - **Collect info from VM monitor**
  - **Record in VM DB**
- **Local monitor**
  - **Go through web port of the clouds**

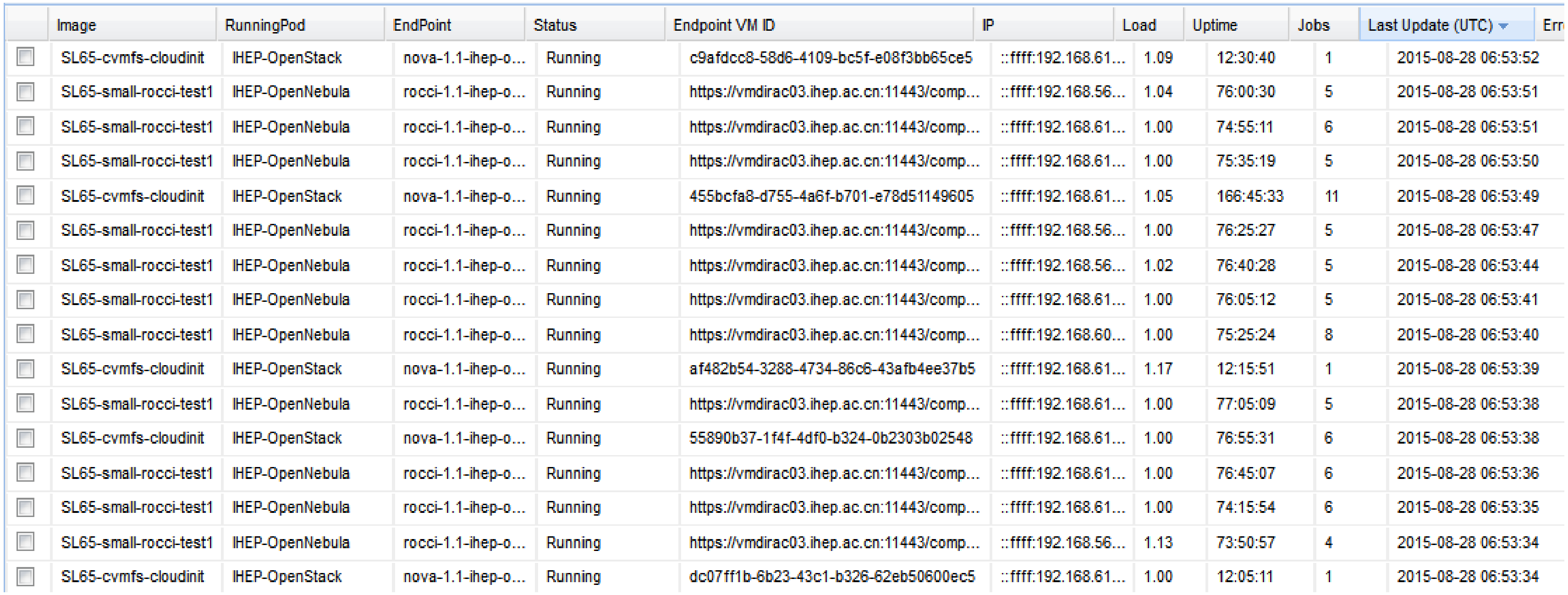
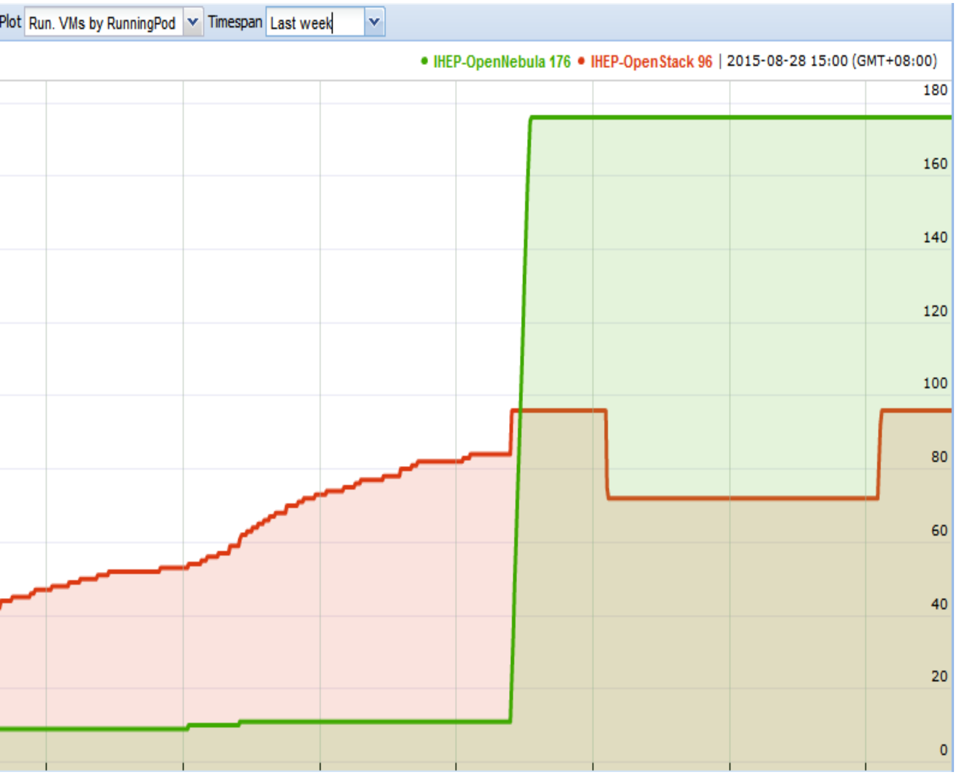


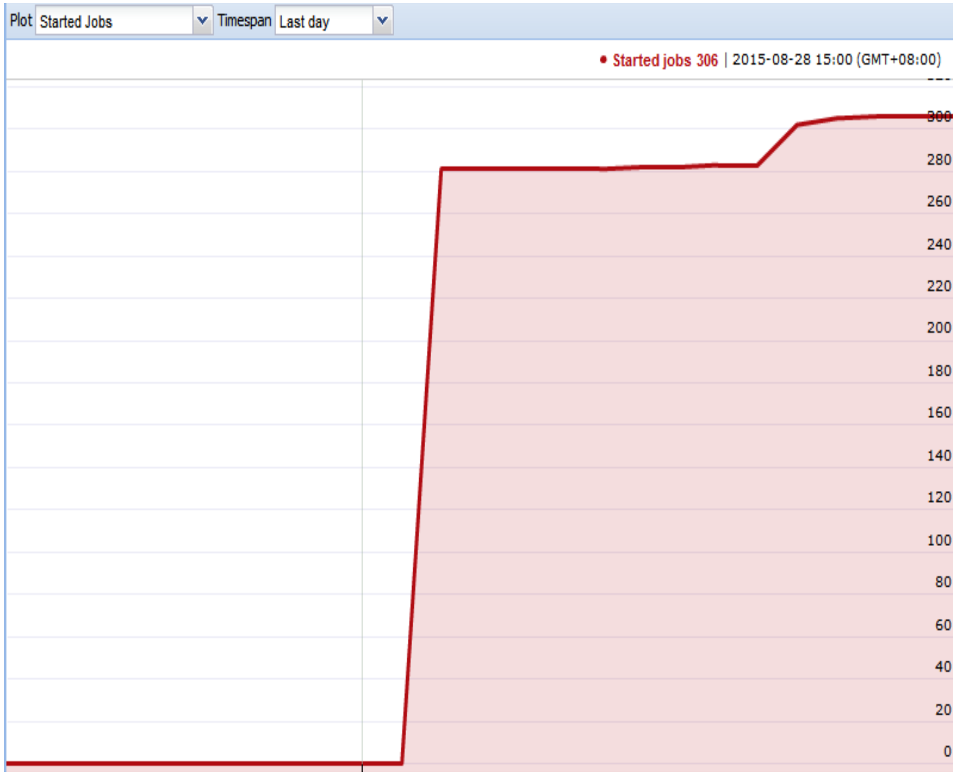
	Image	RunningPod	EndPoint	Status	Endpoint VM ID	IP	Load	Uptime	Jobs	Last Update (UTC)	Err
□	SL65-cvmfs-cloudinit	IHEP-OpenStack	nova-1.1-ihep-o...	Running	c9afdcc8-58d6-4109-bc5f-e08f3bb65ce5	::ffff:192.168.61...	1.09	12:30:40	1	2015-08-28 06:53:52	
□	SL65-small-rocci-test1	IHEP-OpenNebula	rocci-1.1-ihep-o...	Running	https://vmdirac03.ihep.ac.cn:11443/comp...	::ffff:192.168.56...	1.04	76:00:30	5	2015-08-28 06:53:51	
□	SL65-small-rocci-test1	IHEP-OpenNebula	rocci-1.1-ihep-o...	Running	https://vmdirac03.ihep.ac.cn:11443/comp...	::ffff:192.168.61...	1.00	74:55:11	6	2015-08-28 06:53:51	
□	SL65-small-rocci-test1	IHEP-OpenNebula	rocci-1.1-ihep-o...	Running	https://vmdirac03.ihep.ac.cn:11443/comp...	::ffff:192.168.61...	1.00	75:35:19	5	2015-08-28 06:53:50	
□	SL65-cvmfs-cloudinit	IHEP-OpenStack	nova-1.1-ihep-o...	Running	455bcfa8-d755-4a6f-b701-e78d51149605	::ffff:192.168.61...	1.05	166:45:33	11	2015-08-28 06:53:49	
□	SL65-small-rocci-test1	IHEP-OpenNebula	rocci-1.1-ihep-o...	Running	https://vmdirac03.ihep.ac.cn:11443/comp...	::ffff:192.168.56...	1.00	76:25:27	5	2015-08-28 06:53:47	
□	SL65-small-rocci-test1	IHEP-OpenNebula	rocci-1.1-ihep-o...	Running	https://vmdirac03.ihep.ac.cn:11443/comp...	::ffff:192.168.56...	1.02	76:40:28	5	2015-08-28 06:53:44	
□	SL65-small-rocci-test1	IHEP-OpenNebula	rocci-1.1-ihep-o...	Running	https://vmdirac03.ihep.ac.cn:11443/comp...	::ffff:192.168.61...	1.00	76:05:12	5	2015-08-28 06:53:41	
□	SL65-small-rocci-test1	IHEP-OpenNebula	rocci-1.1-ihep-o...	Running	https://vmdirac03.ihep.ac.cn:11443/comp...	::ffff:192.168.60...	1.00	75:25:24	8	2015-08-28 06:53:40	
□	SL65-cvmfs-cloudinit	IHEP-OpenStack	nova-1.1-ihep-o...	Running	af482b54-3288-4734-86c6-43afb4ee37b5	::ffff:192.168.61...	1.17	12:15:51	1	2015-08-28 06:53:39	
□	SL65-small-rocci-test1	IHEP-OpenNebula	rocci-1.1-ihep-o...	Running	https://vmdirac03.ihep.ac.cn:11443/comp...	::ffff:192.168.61...	1.00	77:05:09	5	2015-08-28 06:53:38	
□	SL65-cvmfs-cloudinit	IHEP-OpenStack	nova-1.1-ihep-o...	Running	55890b37-1f4f-4df0-b324-0b2303b02548	::ffff:192.168.61...	1.00	76:55:31	6	2015-08-28 06:53:38	
□	SL65-small-rocci-test1	IHEP-OpenNebula	rocci-1.1-ihep-o...	Running	https://vmdirac03.ihep.ac.cn:11443/comp...	::ffff:192.168.61...	1.00	76:45:07	6	2015-08-28 06:53:36	
□	SL65-small-rocci-test1	IHEP-OpenNebula	rocci-1.1-ihep-o...	Running	https://vmdirac03.ihep.ac.cn:11443/comp...	::ffff:192.168.61...	1.00	74:15:54	6	2015-08-28 06:53:35	
□	SL65-small-rocci-test1	IHEP-OpenNebula	rocci-1.1-ihep-o...	Running	https://vmdirac03.ihep.ac.cn:11443/comp...	::ffff:192.168.56...	1.13	73:50:57	4	2015-08-28 06:53:34	
□	SL65-cvmfs-cloudinit	IHEP-OpenStack	nova-1.1-ihep-o...	Running	dc07ff1b-6b23-43c1-b326-62eb50600ec5	::ffff:192.168.61...	1.00	12:05:11	1	2015-08-28 06:53:34	

VM monitor

- The total number of VMs by RunningPod
- The total jobs run in the Clouds



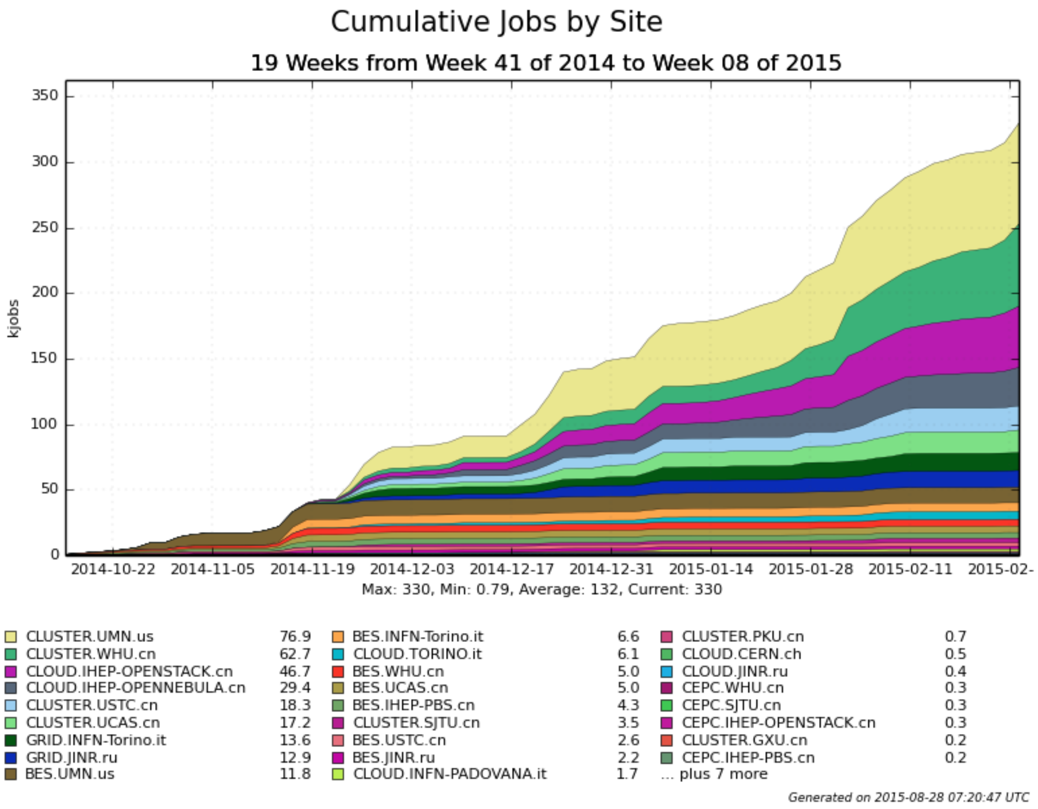
Plot Run. VMs by RunningPod
Timespan Last week



Plot Started Jobs
Timespan Last day

Accounting

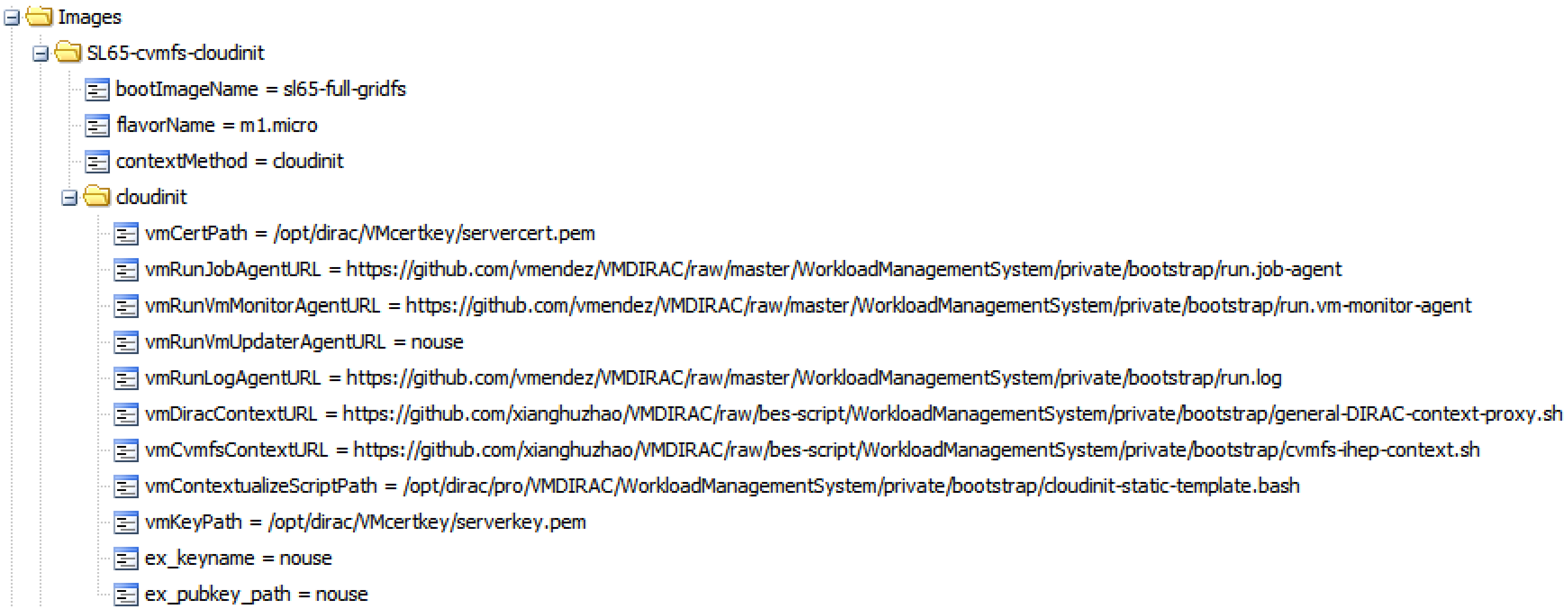
- A history view of cloud as other resources



- Thank you!

“Image” section

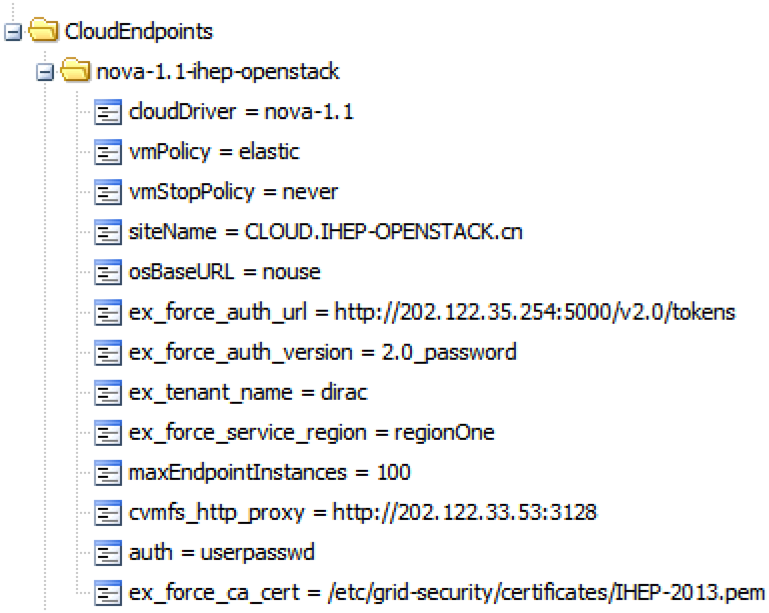
- bootImageName
- FlavorName
- image name containing
  - OS, software……



```
Images
  SL65-cvmfs-cloudinit
    bootImageName = sl65-full-gridfs
    flavorName = m1.micro
    contextMethod = cloudinit
    cloudinit
      vmCertPath = /opt/dirac/VMcertkey/servercert.pem
      vmRunJobAgentURL = https://github.com/vmendez/VMDIRAC/raw/master/WorkloadManagementSystem/private/bootstrap/run.job-agent
      vmRunVmMonitorAgentURL = https://github.com/vmendez/VMDIRAC/raw/master/WorkloadManagementSystem/private/bootstrap/run.vm-monitor-agent
      vmRunVmUpdaterAgentURL = nouse
      vmRunLogAgentURL = https://github.com/vmendez/VMDIRAC/raw/master/WorkloadManagementSystem/private/bootstrap/run.log
      vmDiracContextURL = https://github.com/xianghuzhao/VMDIRAC/raw/bes-script/WorkloadManagementSystem/private/bootstrap/general-DIRAC-context-proxy.sh
      vmCvmfsContextURL = https://github.com/xianghuzhao/VMDIRAC/raw/bes-script/WorkloadManagementSystem/private/bootstrap/cvmfs-ihep-context.sh
      vmContextualizeScriptPath = /opt/dirac/pro/VMDIRAC/WorkloadManagementSystem/private/bootstrap/cloudinit-static-template.bash
      vmKeyPath = /opt/dirac/VMcertkey/serverkey.pem
      ex_keyname = nouse
      ex_pubkey_path = nouse
```

“Endpoint” Section

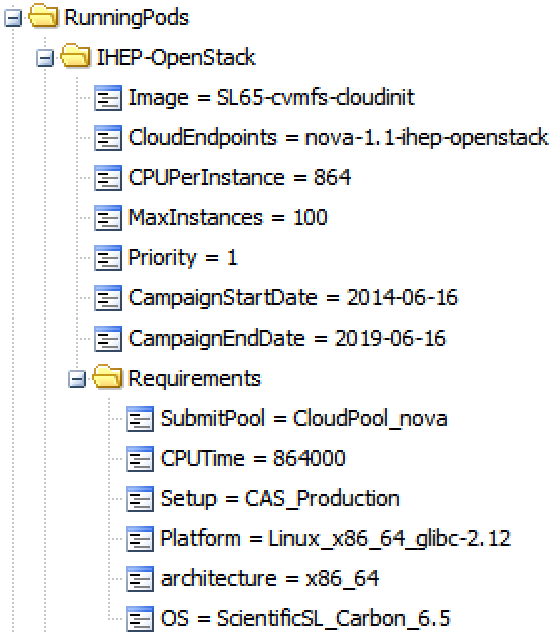
- Necessary info to connect with Cloud
- cloudDriver is the interface to connect cloud
- It is related directly with cloud name known by users



```
CloudEndpoints
  nova-1.1-ihep-openstack
    cloudDriver = nova-1.1
    vmPolicy = elastic
    vmStopPolicy = never
    siteName = CLOUD.IHEP-OPENSTACK.cn
    osBaseURL = nouse
    ex_force_auth_url = http://202.122.35.254:5000/v2.0/tokens
    ex_force_auth_version = 2.0_password
    ex_tenant_name = dirac
    ex_force_service_region = regionOne
    maxEndpointInstances = 100
    cvmfs_http_proxy = http://202.122.33.53:3128
    auth = userpasswd
    ex_force_ca_cert = /etc/grid-security/certificates/IHEP-2013.pem
```

“Running Pod” section

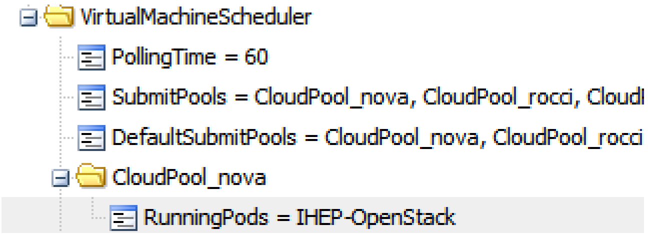
- Requirements define the running env this RunningPods can provide
- Separate image and requirements? If image doesn’t match the requirements?



```
RunningPods
  IHEP-OpenStack
    Image = SL65-cvmfs-cloudinit
    CloudEndpoints = nova-1.1-ihep-openstack
    CPUPerInstance = 864
    MaxInstances = 100
    Priority = 1
    CampaignStartDate = 2014-06-16
    CampaignEndDate = 2019-06-16
    Requirements
      SubmitPool = CloudPool_nova
      CPUTime = 864000
      Setup = CAS_Production
      Platform = Linux_x86_64_glibc-2.12
      architecture = x86_64
      OS = ScientificSL_Carbon_6.5
```

“SubmitPools”

- Define available resources to VM scheduler
- Different RunningPods are put into SubmitPools for VM scheduler



```
VirtualMachineScheduler
    PollingTime = 60
    SubmitPools = CloudPool_nova, CloudPool_rocci, CloudI
    DefaultSubmitPools = CloudPool_nova, CloudPool_rocci
    CloudPool_nova
        RunningPods = IHEP-OpenStack
```