Valutazione di Cloudscheduler

Alessandro Italiano
Giacinto Donvito

CloudScheduler main scope

The cloud scheduler itself relies on a preexisting job submission and scheduling resource (currently, a Condor job scheduler running on the cloud at UVIC), and a number of computational clusters running cloud management software

The aim of the cloud scheduling system is to provide a scheduling system that automates the process of manually customizing an environment to suit scientific computational tasks.

The cloud scheduling system provides an environment in which job environment requirements can be specified simply, and can then be submitted in the same manner by which scientists would submit a computational job to a cluster.

The cloud scheduling system is then responsible for creating an environment in the cloud that will execute a large number of jobs efficiently and correctly.

Altering job priority is a known issue

One can use a scheduling algorithm in either Condor Job Scheduler or Cloud Scheduler but one consequence of the current design is that Cloud Scheduler can impact the scheduling decision made by the Condor Job Scheduler(and vice versa). In the simplest case where Condor use FIFO algorithm it may be possible that job are not run in expected order under certain circumstances

The cloud cluster will follow this sequence when a VM running on the cloud has finished executing its initial job (given to the VM by the job scheduler in the cloud scheduling system).

1. An X_VM finishes its initial job.
2. The cloud scheduler:
. detects that the X_VM is no longer executing a job.
. checks job queues for jobs requiring an X_VM type virtual machine.
This termination case represents a very simple method of achieving VM life-cycle management.The use case for this scenario will evolve with further design efforts.

Other issues

- Resources instantiation are done with the same user

- Batch interface is not a plugin

- Condor oriented

BareMetal provisioning, how it works

Building kernel, ramdisk and image using

Load them into glance

create a new OpenStack flavour

Hardware enrolment

```
nova baremetal-node-create --pm_address=... --pm_user=... --pm_password=... $COMPUTE-HOST-NAME $CPU $RAM $DISK $FIRST-MAC
```

Hardware provisioning

```
nova boot --flavor my-baremetal-flavor --image my-image my-baremetal-node
```

CloudScheduler as batch resource provider