How to install FairShareScheduler in Openstack (v. HAVANA)

This document provides the step-by-step installation and configuration procedure of the FairShareScheduler in Openstack (v.Havana).
Openstack installed by using the tool "packstack" ("rdo-release-havana-8").

1. Installation of the FairShareScheduler

```
$ git clone https://github.com/CloudPadovana/openstack-fairshare-scheduler.git

$ python -m compileall -f
./openstack-fairshare-scheduler/src/fairsharescheduler/
```

for svc in api cert compute conductor scheduler; do service openstack-nova-$svc stop; done

```
$ cp -r openstack-fairshare-scheduler/src/fairsharescheduler/
/usr/lib/python2.6/site-packages/nova/scheduler
```

1.1 Dependencies

```
$ yum install mysql-connector-python
$ wget http://www.webwareforpython.org/downloads/DBUtils/DBUtils-1.1.tar.gz
$ tar xzvf DBUtils-1.1.tar.gz
$ cd DBUtils-1.1
$ python setup.py install
```

2. Creation of the database " scheduler_priority_queue"

Use the script "`openstack-fairshare-scheduler/script`create_db_fairshare_scheduler.sql"

```
 $ mysql -uroot -p <
./openstack-fairshare-scheduler/script/create_db_fairshare_scheduler.sql
```

Authors: Eric Frizziero (INFN-PD), Lisa Zangrando (INFN-PD)

In order to verify the database creation, connect to the mysql shell:

```
mysql> show databases;
+--------------------------+
| Database                 |
+--------------------------+
| information_schema       |
| glance                   |
| keystone                 |
| mysql                    |
| nova                     |
| ovs_neutron              |
| scheduler_priority_queue |
| test                     |
+--------------------------+
8 rows in set (0.00 sec)
```

2.1 Grant permissions to user "root"

Connect to the mysql shell (replace *HOSTNAME* and *psw_root* with the correct values):

```
$ mysql -r root -p

GRANT select on keystone.* to 'root'@'HOSTNAME' IDENTIFIED BY 'psw_root';
GRANT select on nova.* to 'root'@'HOSTNAME IDENTIFIED BY 'psw_root';
GRANT all on scheduler_priority_queue.* to 'root'@'HOSTNAME' IDENTIFIED BY
'psw_root';
FLUSH PRIVILEGES;
```

3. Configuration of the FairShareScheduler

In order to configure the FairShareScheduler, edit/modify the file "/etc/nova/nova.conf" as follows:

```
#
# Options for FairShareScheduler
#

# time-window length (day)
period_length=7

# number of time-windows
num_of_periods=3

# update rate fairShare (minute)
rate=5
```

```
# definition of weights
age_weight=1000
fair_share_vcpus_weight=10000
fair_share_memory_weight=7000
# decay_weight="0.5"

# number of workers which process the users requests in parallel
thread_pool_size=1

# mysql parameters
mysql_host="the_mysql_host_ip"
mysql_scheduler_db="scheduler_priority_queue"
mysql_user="the_mysql_user"
mysql_passwd="the_mysql_password"
mysql_pool_size=10

# definition of the projects and users shares:
# default value for project share.
default_project_share=10

#project_shares={'prjX_name':shareX, … 'prjY_name':shareY}
#project_shares={'p1':44, 'p2':57}

#user_shares={'prjN_name':{'usrX_name':shareX, 'usrY_name':shareY}, … }
#user_shares={'p1':{'p1_u1':11, 'p1_u3':13}, 'p2':{'p2_u1':21, 'p1_u3':13}}

#
# Options defined in nova.scheduler.manager
#

# Default driver to use for the scheduler (string value)
scheduler_driver=nova.scheduler.fairsharescheduler.fairshare_scheduler.FairSha
reScheduler

# the topic scheduler nodes listen on (string value)
scheduler_topic=scheduler (verificare che ci sia)

#
# Options defined in nova.openstack.common.notifier.api
#

# Driver or drivers to handle sending notifications (multi valued)
notification_driver=nova.openstack.common.notifier.rpc_notifier (verificare
che ci sia)

#
# Options defined in nova.openstack.common.notifier.rpc_notifier
```

```
#

# AMQP topic used for OpenStack notifications (list value)
notification_topics=notifications (verificare che ci sia)

notify_on_state_change = vm_and_task_state
```

4. Changes to Openstack code

Please, pay attention to the indentation of the code.

4.1 */usr/lib/python2.6/site-packages/nova/cert/manager.py*

```
class CertManager(manager.Manager):

    def destroy(self):    <- - - Add this method
        pass
```

4.2 */usr/lib/python2.6/site-packages/nova/compute/manager.py*

```
class ComputeManager(manager.SchedulerDependentManager):

    def destroy(self):       <- - - Add this method
        pass
```

4.3. */usr/lib/python2.6/site-packages/nova/conductor/manager.py*

```
class ConductorManager(manager.Manager):

    def destroy(self):       <- - - Add this method
        pass
```

4.4 */usr/lib/python2.6/site-packages/nova/service.py*

```
class Service(service.Service):
    def stop(self):
        LOG.info("service stop")  <- - - Add this line

        try:
            self.conn.close()
         except Exception:
            pass

      super(Service, self).stop()
      self.manager.destroy() <- - - Add this line
```

4.5 */usr/lib/python2.6/site-packages/nova/scheduler/manager.py*

```
class SchedulerManager(manager.Manager):

    def destroy(self):              <- - - Add this method
```

```
        LOG.info("manager destroy")
        self.driver.destroy()
```

4.6 *`/usr/lib/python2.6/site-packages/nova/openstack/common/service.py`*

```
class Launcher(object):

    def launch_service(self, service):
        """Load and start the given service.

        :param service: The service you would like to start.
        :returns: None

        """
        service.backdoor_port = self.backdoor_port
        self._services.add_thread(self.run_service, service)
        self._service = service            <- - - Add this line
        LOG.info("launcher added service") <- - - Add this line

    def stop(self):
        """Stop all services which are currently running.

        :returns: None

        """
        LOG.info("launcher stopping all services")

        self._services.stop()
        self._service.stop()    <- - - Add this line
```

4.6 Compile the code

`python -m compileall -f /usr/lib/python2.6/site-packages/nova`

5. Restart Openstack services

for svc in api cert compute conductor scheduler; do service openstack-nova-$svc start; done