Hands-on activities

Activity 17

Creating a GPU queue for nodes with NVIDIA cards

Requirements

- Working ROCKS cluster with masternode and at least one compute node (possibly Vmware or VirtualBox)

Steps

- Open a terminal window or remotely login to the cluster master node using the ssh command
- Become the root user using the command
  - **su -**
- Optionally specify your favourite Linux editor
  - **export EDITOR=nano**
- Create new OR modify existing queue
  - **qconf –aq gpu** *OR* **qconf –mq gpu**
  - *This opens an editor screen various options*
  - Make the following changes:
    - Set desired queuename
      queuename **gpu**
    - Specify which computers can run jobs on this queue, separate entries using space
      hostlist **compute-0-13.local**
    - Specify how may GPU devices are available on each node, list is comma separated and each enclosed in []
      slots 1**,[compute-0-13.local=1]**
    - Set maximum runtine allowed to
      h_rt **12:00:00**
    - Exit editor with saving of file
- Create a complex selector for the queue
  - **qconf -mc**
  - *This opens an editor screen various options*
  - Add the following line if it does not exist already:

    gpu gpu INT <= YES YES 0 0
  - *Exit the Editor saving the file*

- Modify the node information to enforce the selector
  - **qconf –me compute-0-13**
  - *This opens an editor screen various options*
  - Make the following changes:

        - Set desired complex_values

          complex_values **gpu=1**
        - Exit editor with saving of file
- Optionally remove the node from the general queue
    - **qconf –mq all.q**
    - *This opens an editor screen various options*
    - Make the following changes:
        - Set the slots available from the node to 0

          slots 1**,[compute-0-13.local=0]**
        - Exit editor with saving of file

Activity 18

Creating a MIC queue for nodes with Intel Xeon Phi cards

Requirements

- Working ROCKS cluster with masternode and at least one compute node (possibly Vmware or VirtualBox)

Steps

- Open a terminal window or remotely login to the cluster master node using the ssh command
- Become the root user using the command
  - **su -**
- Optionally specify your favourite Linux editor
  - **export EDITOR=nano**
- Create new OR modify existing queue
  - **qconf –aq mic** *OR* **qconf –mq mic**
  - *This opens an editor screen various options*
  - Make the following changes:
    - Set desired queuename

      queuename **mic**
    - Specify which computers can run jobs on this queue, separate entries using space

      hostlist **compute-0-12.local**
    - Specify how may GPU devices are available on each node, list is comma separated and each enclosed in []

      slots 1**,[compute-0-12.local=1]**
    - Set maximum runtime allowed to

      h_rt **12:00:00**
    - Exit editor with saving of file
- Create a complex selector for the queue
  - **qconf -mc**
  - *This opens an editor screen various options*
  - Add the following line if it does not exist already:

    mic mic INT <= YES YES 0 0
  - *Exit the Editor saving the file*

- Modify the node information to enforce the selector
  - **qconf –me compute-0-12**
  - *This opens an editor screen various options*
  - Make the following changes:
    - Set desired complex_values

complex_values **mic=1**

- Exit editor with saving of file

- Optionally remove the node from the general queue
  - **qconf –mq all.q**
  - *This opens an editor screen various options*
  - Make the following changes:
    - Set the slots available from the node to 0

      slots 1**,[compute-0-12.local=0]**
    - Exit editor with saving of file

Activity 19

Re-installing a compute node

Requirements

- Working ROCKS cluster with masternode and at least one compute node (possibly Vmware or VirtualBox)

Steps

- Open a terminal window or remotely login to the cluster master node using the ssh command
- Become the root user using the command
  - **su -**
- Flag the compute node to be reinstall
  - **rocks set host boot *{compute-0-0}* action=install**
- Force a reboot of the node if it is not running any jobs free
  - **rocks run host *{compute-0-0}* reboot**

Activity 20

Managing nodes

Deploying additional packages to nodes and also automatically configuring nodes for GPU computing

Requirements

- Working ROCKS cluster with masternode and at least one compute node (possibly Vmware or VirtualBox)

Steps

- Open a terminal window or remotely login to the cluster master node using the ssh command
- Become the root user using the command
  - **su -**
- **Approach 1:** Running the command immediate on a target node or use the special host-name **compute** for ALL hosts ; requires the target node(s) to be up and running
  - Install an RPM application
    - **rocks run host *{hostname|compute}* "yum -y install kernel-devel"**
  - Reboot **a** nodes
    - **rocks run host *{hostname|compute}* "reboot"**
  - Power off **a** nodes
    - **rocks run host *{hostname|compute}* "poweroff"**

- **Approach 2:** Modifying the Kick-start installation process. ROCKS provides an XML file for setting up additional packages to be deployed
  - Change directory to the right location
    - **cd /export/rocks/install/site-profiles/7.0/nodes**
  - **Optionally** create a new extend-base.xml ONLY if it does not exist already
    - **cp skeleton.xml extend-base.xml**
  - Modify the extend-base.xml
    - **nano extend-base.xml**
  - Identify the section for adding new packages and add new entries by adding new entries like below, please replace kernel-devel with the package name of your choice.
    - **<package>*kernel-devel*</package>**
  - This file can also be used to perform some configuration tasks after the re-install activity by modifying the POST section
    - **<post></post>**
  - Save the file
  - Check the XML syntax of the file
    - **xmllint -noout extend-base.xml**

- Rebuild the ROCKS information about packages
  - **cd /export/rocks/install**
  - **rocks create distro**
- Re-install the nodes as discussed in activity 19.

References

http://central-7-0-x86-64.rocksclusters.org/roll-documentation/base/7.0/customization.html

Activity 21

Additional notes for ROCKS + SGE clusters

Switching the SGE type of a compute node to execution or submit should be carried out using the appropriate rocks command followed by a node re-install.

SGE does many features such as user or group accounting and other advanced features. For example SGE hostgroups may be used to easily implement the tourque node feature+maui class.

Restricting direct ssh logins to compute nodes requires the use of epilog and prolog scripts which have to be setup and tested.

Switching CPU states to low-power when idle could result in some energy saving, however this requires proper implementation using the prolog and epilog scripts.

SGE provides a comprehensive Linux GUI tool that may be useful for users. However in order to use it over an ssh connection from your Linux Laptops, it may be necessary to perform the following steps

- Install the xfonts-75dpi package using the right command such as

  sudo apt-get install xfonts-75dpi

- Then update the fonts

  xset +fp /usr/share/fonts/X11/75dpi

  xset fp rehash

- Now login to ROCKS login or frontend node and run the command

  qmon