ASIGNATURA: Computación de altas prestaciones

Proyecto básico: Ejecución y Planificación de tareas en cluster Rocks

El objetivo es aprender a ejecutar tareas en un cluster rocks y utilizar los gestores de colas para la planificación de tareas.

Los videos de apoyo que explican el proceso de instalación siguiendo este guión son:

1.12 Cluster Rocks- 5.4.1. Creación, sincronización de usuarios y reinstalación de nodos.mp4
1.13 Cluster Rocks- 5.4.2. Instalación de aplicaciones.mp4
1.14 Cluster Rocks- 5.4.3. Ejecución MPI.mp4
1.15 Cluster Rocks- 5.4.4. Lanzar tareas a colas.mp4
1.16 Cluster Rocks- 5.5.1. Gestor de colas SGE.mp4
1.17 Cluster Rocks- 5.5.2. Creación de un Script para SGE.mp4
1.18 Cluster Rocks- 5.5.3. Gestión de tareas a las colas.mp4
1.19 Cluster Rocks- 5.5.4. Gestión de colas de ejecución.mp4
1.20 Cluster Rocks- 5.5.5. Creación de colas personalizadas en el cluster.mp4

1.- Instalación de aplicaciones en los nodos de computo

Ver el directorio compartido por todos los nodos de computo.

En el frontend comprobar el contenido del siguiente directorio

```
# cd /export/apps
```

y añada los ficheros que se desean compartir por el resto de nodos. Por ejemplo el fichero tmphora y estará disponible en los nodos de computo en el directorio **/share/apps**

Pruebe los siguientes comandos (desde el usuario root)./con --:

```
# cd /export/apps
# touch tmpnombreyhora
# hostname >> tmpnombreyhora
# date >> tmpnombreyhora
# ssh compute-0-0
# cd /share/apps
# cat  tmpnombreyhora
```

INSTALAR HWLOC (Hardware Locality)
https://www.open-mpi.org/projects/hwloc/

Descargar el código fuente del programa:
https://download.open-mpi.org/release/hwloc/v2.0/hwloc-2.0.3.tar.gz

Crear un directoriopwd
Hwloc/opt

mkdir /export/apps/tools

Descomprimir en /export/apps/tools
#tar xvf hwloc-2.0.3.tar.gz

Entre en el directorio que se ha creado al descomprimir (hwloc-2.0.3) y los pasos para configurar y compilar con:

./configure --prefix=/export/apps/tools
make
make install
y actualice las variables de entorno
```
export PATH=$PATH:/share/apps/tools/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/share/apps/tools/lib
```

Pruebe el funcionamiento ejecutando desde el frontend
hwloc-info
lstopo

Pruebe el funcionamiento ejecutando desde los nodos compute-0-x y explique las diferencias.

ssh compute-0-0
```
#export PATH=$PATH:/share/apps/tools/bin
#export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/share/apps/tools/lib
```

lstopo

2.- Ejecución en varios nodos con MPI

En el frontend con un usuario diferente de root, comprobar que se puede ejecutar el siguiente comando:

$ /opt/openmpi/bin/mpirun -np 10 hostname

El comando mpirun permite ejecutar un comando, en este caso el comando hostname las veces indicadas en el parámetro –np (número de procesos).

Crear ficheros de maquinas para ejecutar MPI

Con un editor cree el siguiente fichero y denomínelo **maquinasMPI.txt**
compute-0-0
compute-0-1
compute-0-2

Pruebe el siguiente comando

$ /opt/openmpi/bin/mpirun -np 10 –machinefile maquinasMPI.txt hostname

y explique como varía el resultado respecto al comando anterior, respondiendo a las siguientes preguntas:

- ¿Se ha ejecutado algún proceso en el frontend?
- ¿Cómo se puede ejecutar parte de los procesos en el frontend?
- Varíe el número de procesos e intente deducir el reparto de tareas que se utiliza.

Ejecutar programas de test realizados con MPI

Pruebe y explique el funcionamiento de los siguientes programas.

$ /opt/openmpi/bin/mpirun -np 6 -machinefile maquinasMPI.txt /opt/mpi-tests/bin/mpi-ring

$ /opt/openmpi/bin/mpirun -np 6 -machinefile maquinasMPI.txt /opt/mpi-tests/bin/mpi-verify

3.- Utilizar el gestor de colas SGE

En Rocks el manejador de colas por defecto es Grid Engine (SGE, incluido en el roll sge),
Se puede probar que el SGE de nuestro cluster funciona, enviando una serie de trabajos sencillos al manejador SGE mediante el comando qsub.

Comandos básicos del gestor de colas

- qsub – Submission script. This submits user Jobs to the queuing system.
- qstat – Reporting script. This reports the status of the queues on the system.
- qdel – Deletion script. This deletes a running job from the queuing system.

Ejemplo para comprobar el estado de la cola

```
$ qstat -f
```

Cómo probar el sistema de colas para distribuir tareas a los nodos

Probar que el SGE del cluster funciona, enviando una serie de trabajos sencillos al manejador SGE mediante el comando qsub.

3.1.- Crear el fichero para enviar el trabajo al gestor de colas

Para ello, primero editamos este código y lo guardamos en un archivo llamado por ejemplo testSGE cuyo contenido es:

```
#!/bin/bash

#$ -j y
#$ -S /bin/bash
#$ -cwd

hostname
date
echo “comando ps  inicial”
ps
sleep 10; date; echo “comando ps final”;ps
echo “Fin”
```

Permitir la ejecución de este script y ejecutar con:

```
$ chmod +x  testSGE
$ ./testSGE
```

comprobar que funciona.
No se ha ejecutado en ninguna cola y la salida se puede ver en la terminal

3.2.- Lanzar el script a la cola de ejecución:

Ejecutar el comando:

$ qsub testSGE

3.3. - Comprobar el estado de la cola

Ejecutar el comando:

$ qstat

vea la diferencia al usar

$ qstat –f

Cuando haya terminado puede ver la salida o los posibles errores en los fiheros creados con idéntico nombre a la tarea y extensiones .o y .e en el directorio actual.

$ cat testSGE.o<número>

Pruebe a lanzarlo varias veces, por ejemplo con:

$ for ((i=1;i<=10;i+=1)); do qsub testSGE; done

Compruebe el estado de la cola y deduzca como se están repartiendo los trabajos.

3.4 Otro ejemplo con script Perl

Otra posibilidad es utilizar script de perl, primero editamos este código y lo guardamos en un archivo llamado por ejemplo testSGE.pl :

```
#!/usr/bin/perl
for(my $i=0;$i<30;$i++){ sleep(2) }
print "ok!\n";
```

Hay que hacer el archivo ejecutable con
$ chmod +x testSGE.pl

Para enviar varias instancias de este programita al cluster y comprobar que los trabajos efectivamente se ejecutan haciendo:

```
$ for ((i=1;i<=10;i+=1)); do qsub testSGE.pl; done

Your job 2 ("testSGE.pl") has been submitted
Your job 3 ("testSGE.pl") has been submitted
Your job 4 ("testSGE.pl") has been submitted
Your job 5 ("testSGE.pl") has been submitted
Your job 6 ("testSGE.pl") has been submitted
Your job 7 ("testSGE.pl") has been submitted
Your job 8 ("testSGE.pl") has been submitted
Your job 9 ("testSGE.pl") has been submitted
Your job 10 ("testSGE.pl") has been submitted
Your job 11 ("testSGE.pl") has been submitted
```

Para ver el estado de las colas hay que ejecutar:
$ qstat –f

```
queuename qtype resv/used/tot. load_avg arch states
---------------------------------------------------------------------------
all.q@compute-0-0.local BIP 0/0/1 0.05 lx26-x86
############################################################################
 - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS
############################################################################
 3 0.55500 testSGE.pl pepe qw 06/10/2011 09:41:45 1
 4 0.55500 testSGE.pl pepe qw 06/10/2011 09:41:45 1
 5 0.55500 testSGE.pl pepe qw 06/10/2011 09:41:45 1
 ….
```

4.- Gestionar colas

Ver la configuración actual de las colas SGE

Colas disponibles:

$ qconf -sql

Entornos paralelos.

$ qconf -spl

Grupos de hosts.

$ qconf -shgrpl

Crear un grupo de equipos

Un grupo de equipos se define con un nombre y un listado de nodos.
Por defecto el grupo @allhosts contiene todos los nodos.

$ qconf -shgrp @allhosts

Creamos nuestro propio grupo de equipos de la siguiente forma:

$ qconf -shgrp @allhosts > /nuestra/ruta/ejemplo/a/MyHostGroup.txt

Se edita el campo group name (por ejemplo, llamamos a este nuevo grupo @MyHostGroup), y lo añadimos a los gestionados por SGE) con los siguientes comandos:

El proceso de añadir el grupo de host necesita ser root:
$ su

# qconf -Ahgrp /nuestra/ruta/ejemplo/a/MyHostGroup.txt
# exit
$ qconf -shgrpl

Crear y configurar una cola de trabajo alternativa a la cola all.q

La cola all.q se proporciona por defecto y contiene siempre a todos los nodos del cluster.
Puede interesar crear colas con un subconjunto de los nodos que compartan ciertas características, o bien modificar cualquiera de las muchas opciones de la cola de trabajo.

El proceso sería siendo root:

1. Ver todas las colas de trabajo gestionadas por SGE,
2. Mostrar en detalle la configuración de una cola en concreto
3. Copiar la configuración de la cola en un archivo.

Esto se realiza con los siguientes comandos:

$ qconf -sql
$ qconf -sq all.q

$ qconf -sq all.q >/nuestra/ruta/ejemplo/MyCola.txt

Se debe modificar con un editor de textos el campo qname por otro nombre que no sea all.q, por ejemplo MyCola.q y también modificar los campos

- hostlist con el grupo de nodos que corresponda
- slots para que sea coherente con el campo hostlist.

4. Después se añade a las colas gestionadas por SGE y se modifican el resto de campos que se deseen con los dos siguientes comandos:

qconf -Aq / nuestra/ruta/ejemplo/MyCola.txt

y se puede modificar con:
$ qconf -mq MyCola.q

5.- Cómo enviar tareas MPI al gestor de colas

En primer lugar como usuario diferente de root, compile los ejemplos de mpi

```
$ cd $HOME
$ mkdir test
$ cd test
$ cp /opt/mpi-tests/src/*.c .
$ cp /opt/mpi-tests/src/Makefile .
$ make
```

Prepare un script para enviar las tareas a la cola SGE. Con un editor escriba las siguientes líneas en un fichero de texto y denomínelo mpiring.qsub

```
#!/bin/bash
#
#$ -cwd
#$ -j y
#$ -S /bin/bash
#

/opt/openmpi/bin/mpirun $HOME/test/mpi-ring
```

Para enviar el job mpi-ring.qsub con ejecución paralela se hace de manera similar pero es necesario usar las opciones :

```
-pe orte N
```

Donde N indica el número de proceso MPI que se lanzan.

Por ejemplo para enviar el job con 2 procesos:

```
$ qsub -pe orte 2 mpi-ring.qsub
```

Cuando finalize el job la salida estará en el fichero `mpi-ring.qsub.o*`. y los mensajes de error en `mpi-ring.qsub.po*`.

Ejercicio: Prueba a lanzarlo con 16 procesos y explique como funciona

```
$ qsub -pe orte 16 mpi-ring.qsub
```

Cómo borrar procesos de la cola

Utilizando

qdel con el job id.

Por ejemplo para borrar el job denominado fluent.sh en la cola SGE:

```
[sysadm1@frontend-0 sysadm1]$ qsub fluent.sh your job 31 ("fluent.sh") has been submitted

$ qstat
job-ID  prior name       user         state submit/start at     queue      master  ja-task-ID
------------------------------------------------------------------------------------------------
31     0 fluent.sh  sysadm1      t     12/24/2003 01:10:28 comp-pvfs- MASTER

$ qdel 31
sysadm1 has registered the job 31 for deletion
$ qstat
$
```

EJERCICIOS: Práctica 1 – Parte0 – Sección2

Ejercicio 6: Con un editor cree un fichero con el siguiente contenido

```
compute-0-0
compute-0-1
compute-0-2
```

y denomínelo **maquinasMPI.txt**

Pruebe el siguiente comando:

```
$ /opt/openmpi/bin/mpirun -np 10 –machinefile maquinasMPI.txt  hostname
```

y explique como varía el resultado respecto al comando:

```
/opt/openmpi/bin/mpirun -np 10 hostname
```

Responda a las siguientes preguntas:

- ¿Se ha ejecutado algún proceso en el frontend?
- ¿Cómo se puede ejecutar parte de los procesos en el frontend?
- Varíe el número de procesos e intente deducir el reparto de tareas que se utiliza.

Ejercicio 7: Compile los ejemplos de mpi disponibles en /opt/mpi-tests/e indique como funcionan.

Ejercicio 8: Realizar una cola denominada colapares.q basándose en la cola all.q que tenga solo los nodos con nombres compute-0-x, siendo x par. Compruebe su funcionamiento.

Ejercicio 9: Cree nuevas colas de acuerdo a criterios que le parezcan significativos, por ejemplo cola1core para máquinas con un solo procesador y cola2core para máquinas con dos procesadores. Para ello reinstale el nodo 1 con 2 cores y cree una nuevo nodo compute-0-3 con 2 cores.

Realice pruebas para comprobar el funcionamiento de las colas creadas.