Clúster ctcomp2

Diego Rodríguez Martínez

Centro Singular de Investigación
en Tecnoloxías da Información – CITIUS

Universidade de Santiago de Compostela

citius.usc.es

Índice

Índice

Clúster de computación

Definición

- Conjunto de **nodos computacionales** . . .
- . . . interconectados mediante una **red dedicada** . . .
- . . . y que pueden actuar como un **único** nodo computacional

En la práctica, esto se traduce en:

- **potencia** computacional. . .
  - Ejecución de un trabajo paralelo muy grande
  - Muchas ejecuciones *pequeñas* concurrentes
- . . . **compartida** entre varios usuarios

Clúster de computación

Definición

- Conjunto de **nodos computacionales** . . .
- . . . interconectados mediante una **red dedicada** . . .
- . . . y que pueden actuar como un **único** nodo computacional

En la práctica, esto se traduce en:

- **potencia** computacional. . .
  - Ejecución de un trabajo paralelo muy grande
  - Muchas ejecuciones *pequeñas* concurrentes
- . . . **compartida** entre varios usuarios

Gestión de trabajos en un clúster

Sistema de gestión de colas

- Planificación de los **trabajos** de los usuarios, i.e., gestión de los recursos
  - Trabajo: una ejecución de un programa
  - Cada trabajo puede gestionarse como una **tarea autónoma**
- Minimizar costes y maximizar el rendimiento

Gestión de trabajos en un clúster

Dinámica

1. Los usuarios **envían** al gestor de colas un **script** con el trabajo:
   - Con los comandos necesarios para **ejecutar offline** el programa...
   - ...incluyendo una lista de requisitos
2. Se **registra** el trabajo en una cola
3. El gestor de colas se encarga de **ejecutar** el script en los nodos computacionales
   - cuando estén **disponibles los requisitos** solicitados,...
   - ...y en función de las **prioridades** establecidas

Gestión de trabajos en un clúster

Dinámica

1. Los usuarios **envían** al gestor de colas un **script** con el trabajo:
   - Con los comandos necesarios para **ejecutar offline** el programa...
   - ...incluyendo una lista de requisitos
2. Se **registra** el trabajo en una cola
3. El gestor de colas se encarga de **ejecutar** el script en los nodos computacionales
   - cuando estén **disponibles los requisitos** solicitados,...
   - ...y en función de las **prioridades** establecidas

Gestión de trabajos en un clúster

Dinámica

1. Los usuarios **envían** al gestor de colas un **script** con el trabajo:
   - Con los comandos necesarios para **ejecutar offline** el programa...
   - ...incluyendo una lista de requisitos
2. Se **registra** el trabajo en una cola
3. El gestor de colas se encarga de **ejecutar** el script en los nodos computacionales
   - cuando estén **disponibles los requisitos** solicitados,...
   - ...y en función de las **prioridades** establecidas

Clúster de computación ctcomp2

Características (actuales)

- 7×Nodos computacionales
  - Servidores Blade HP Proliant BL685c G7
  - 4×AMD Opteron 6262HE (16 cores, 1.6 GHz, 16 MB L3)
  - 16×8 GB RAM
- 2×Gigabit Ethernet
- Sistema de ficheros compartido (iSCSI, ∼ 8 TB)
  - `/home/local`: *HOME* de los usuarios
  - `/sfs/`: sistema de ficheros distribuido
- Gestor de colas PBS/Torque

Guía de usuario en http://wiki.citius.usc.es

Índice

Acceso al frontend

Frontend

- Sistema al que se conectan los usuarios para...
  - ...**enviar los trabajos** al sistema de colas
  - ...compilar (si es necesario) el código
  - ...manejar los ficheros de entrada/resultados

  ...

- Los directorios *HOME* y `/sfs` están accesibles en el frontend
  - `/sfs` es el lugar adecuado para situar ficheros muy grandes
- Los usuarios no tienen acceso directo a los nodos
  - Todas las ejecuciones deben hacerse a través del sistema de colas
- Los usuarios **no deben ejecutar** trabajos en el *frontend*
  1. No está preparado para ello (procesador con dos núcleos)
  2. Entorpecería al resto de usuarios conectados

Acceso al frontend

Frontend

- Sistema al que se conectan los usuarios para...
  - ...**enviar los trabajos** al sistema de colas
  - ...compilar (si es necesario) el código
  - ...manejar los ficheros de entrada/resultados

  ...

- Los directorios *HOME* y `/sfs` están accesibles en el frontend
  - `/sfs` es el lugar adecuado para situar ficheros muy grandes
- Los usuarios no tienen acceso directo a los nodos
  - Todas las ejecuciones deben hacerse a través del sistema de colas
- Los usuarios **no deben ejecutar** trabajos en el *frontend*
  1. No está preparado para ello (procesador con dos núcleos)
  2. Entorpecería al resto de usuarios conectados

Acceso al frontend (ssh)

```
me@local$ ssh nome.apelido@ctcomp2.inv.usc.es
Password:
[...]

-----------------------------------------------
>> Clúster CTCOMP2 <<
-----------------------------------------------

Guía de usuario:
http://wiki.citius.usc.es/sysadm:servizos:clust...

-----------------------------------------------
>> diego.rodriguez@usc.es  (Ext.: 16411) <<
-----------------------------------------------
[...]
me@ctcomp2$
```

- PuTTY en Windows...

Acceso al frontend (scp)

- Subir un fichero/directorio hacia ctcomp2:

```
me@local$ scp -P1301 /datos/work/un.fich          \
  nome.apelido@ctXXX.inv.usc.es:~/work/
me@local$ scp -r -P1301 /datos/work/dir/          \
  nome.apelido@ctXXX.inv.usc.es:~/work/
```

- Traer un fichero/directorio desde ctcomp2:

```
me@ctcomp2$ scp -P1301\
  nome.apel@ctcomp2.inv.usc.es:~/un.fich ~/work/
me@ctcomp2$ scp -r -P1301\
  nome.apel@ctcomp2.inv.usc.es:~/dir/ ~/work/
```

- . . . o a través del nautilus, WinSCP, etc.

Índice

modules

- Modificación dinámica de las variables de entorno del usuario
- Gestión *modular* de los *paths*: `PATH`, `LD_LIBRARY_PATH`,...
- El usuario selecciona los programas/librerías que desea utilizar
  - Disponibilidad ad hoc de programas/librerías incompatibles en un mismo sistema (incluyendo versiones)
  - Personalización de los trabajos

```
$ module avail
$ module load jdk
...
$ module list
$ module load openmpi
...
$ module purge
```

PBS/TORQUE

- Envío de un trabajo:

```
ct$ vi script.sh
ct$ chmod u+x script.sh
ct$ qsub script.sh
```

  - **¡¡Permisos de ejecución!!**

- Monitorización de trabajos:

```
ct$ qstat -a
ct$ qdel XXX.ctcomp2.work
ct$ pbsnodes
```

PBS/TORQUE

Ejemplo de script (básico)

```
#!/bin/bash
#PBS -l nodes=1:ppn=64,walltime=1:00:00
#PBS -N ej-java
cd $PBS_O_WORKDIR
module load jdk
java -jar programa.jar
```

- No sabemos cuando terminará (ni cuando empieza) la ejecución del trabajo...
  - Incluso, podrían estar apagados: no se podrá ejecutar el trabajo hasta que se enciendan los recursos solicitados
- ...hay que estar pendiente de la cola (`qstat`, `pbsnodes`)

PBS/TORQUE

Ejemplo de script (mejorado)

```
#!/bin/bash
#PBS -l nodes=1:ppn=64,walltime=5:00:00
#PBS -m ae -M nome.apelido@usc.es
#PBS -N param-java
cd $PBS_O_WORKDIR
module load jdk
echo date
OUTDIR=/sfs/`whoami`/$PBS_JOBNAME
for i in 1 2 3 4 5;
do
  echo date
  java -jar programa.jar $i > /scratch/programa.out
  echo date
  cp /scratch/programa.out $OUTDIR/output_$i.txt
done
```

¿Dudas?

diego.rodriguez@usc.es

Buenas prácticas

- Reservar el número adecuado de núcleos computacionales
- Verificar que estamos ejecutando el número corecto de procesos/hilos
- ¿Necesitamos exclusividad (medidas rendimiento, etc.)? Reserva de un nodo completo
  - Estamos desaprovechando el potencial. . . ¿Es necesario?
  - Si compartimos un nodo, es probable que nuestros trabajos vayan un poco más lentos. . .
  - . . . pero el rendimiento global del sistema es mayor (podemos ahorrar un nodo, por ejemplo)
- No es posible monitorizar un trabajo *batch*: redirigir la salida hacia un sistema de ficheros compartido.

Uso de los sistemas de ficheros compartidos

- `/home/local/nome.apelido/`
  - `$HOME` del usuario (no hay backups)
  - accesible en todos los nodos del clúster + frontend.
  - directorio de referencia en las ejecuciones en los nodos
- `/sfs/`
  - compartido entre todos los nodos de computación + frontend
  - espacio auxiliar durante la ejecución de trabajos
  - no se garantiza la conservación permanente de los ficheros que no hayan sido accedidos en los últimos 30 días
  - Recomendación: nombres *personalizados* para evitar conflictos
- `/scratch/`
  - directorio local (no es adecuado para guardar resultados)
  - almacenamiento temporal local durante la ejecución de una tarea: no es visible desde el resto de nodos ni desde el frontend.
  - Una vez terminado el trabajo, se borra el scratch.
  - Recomendación: nombres *personalizados* para evitar conflictos

¿Otras necesidades?

- Ejecución de **entornos *ad hoc*** con máquinas virtuales (KVM/Qemu)
- **Exclusividad temporal**: es posible sacar *ex professo* nodos de la cola (solicitándolo adecuadamente...)
- ...

¡Nos **adaptaremos** en función de la demanda!

¿Otras necesidades?

- Ejecución de **entornos *ad hoc*** con máquinas virtuales (KVM/Qemu)
- **Exclusividad temporal**: es posible sacar *ex professo* nodos de la cola (solicitándolo adecuadamente...)
- ...

¡Nos **adaptaremos** en función de la demanda!