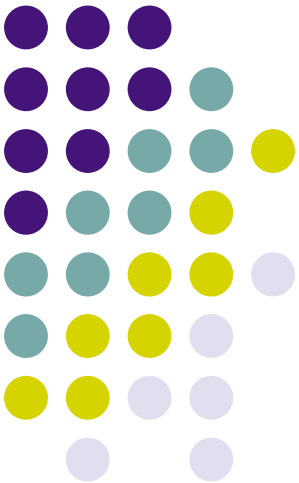
Curso de Computación Científica en Clusters
Administración de Sistemas Operativos III

Luis P. García González
Servicio de Apoyo a la Investigación Tecnológica

Universidad Politécnica de Cartagena



Introducción

- Sistema de gestión de trabajos y recursos de computación

  - Herramienta para que el usuario aproveche las capacidades del sistema de computación.

  - Herramienta para que el administrador del sistema pueda asegurar un uso equitativo del sistema (cumplimiento de las políticas de uso).

Introducción

¿Qué hace un sistema de gestión de trabajos?

- **PONER EN COLA:** tareas para ser ejecutadas en un computador. El usuario envía tareas o “trabajos” al gestor de recursos, y éstos son puestos en cola hasta que el sistema esté preparado para ejecutarlos.
- **PLANIFICAR:** seleccionar que trabajos se van a ejecutar, cuando y donde, conforme a una determinada política con el objetivo de maximizar los recursos (tiempo de computación y tiempo de los usuarios).
- **MONITORIZAR:** realizar un seguimiento de la utilización de los recursos del sistema y asegurar que se están aplicando las políticas de uso.

Dominio público:

- **Torque/Maui:** Teraescale Open-Source Resource and QUEue Manager. Proyecto que surge a partir de OpenPBS (desarrollado por la NASA en los 90) http://www.clusterresources.com

- **Sun Grid Engine:** Proyecto de la empresa SUN destinado a entorno distribuidos y heterogeneos http://gridengine.sunsource.net/

- **Condor:** Proyecto que se centra en la utilización eficiente de recursos computacionales no dedicados. http://www.cs.wisc.edu/condor/description.html

- **SLRUM:** Simple Linux Utility for Resource Management. Proyecto GNU de fácil instalación, con un diseño modular que permite ampliar su funcionalidad a través de su API y plug-ins. https://computing.llnl.gov/linux/slurm/

Comerciales:

- **Platform LSF**: Load Shared Facility. El sistema de gestión de trabajos para entornos de computación de alto rendimiento lider en el mercado. (5 millones de CPUs están siendo gestionadas por este producto).
  http://www.platform.com/workload-management/high-performance-computing

- **Altair PBSPro:** Versión comercial y mejorada del proyecto OpenPBS.
  http://www.pbsworks.com/

- **MOAB:** Versión comercial y mejorada del proyecto MAUI.
  http://www.clusterresources.com

- **IBM LoadLeveler:** http://www-03.ibm.com/systems/software/loadleveler/index.html

Instalaciones de SuperComputación Españolas. Gestores de Recursos

- **Centro de SuperComputación de Murcia:** 1 nodo de 128 cores Intel Itanium-2 y 102 nodos Intel Xeon (816 cores). Load Shared Facility.
- **Centro de SuperComputación de Castilla y León:** 304 nodos Intel Xeon Quad Core (2432 cores). Sun Grid Engine.
- **Centro Nacional de Supercomputación (BSC):** 2560 nodos de IBM PowerPC 970 (10240 procesadores) y 1 nodo de 256 cores Intel Itanium-2. SLRUM+MOAB.
- **Centro de SuperComputación de Extremadura:** 2 nodos de 128 cores cada uno Intel Itanium-2. Load Shared Facility.
- **Centro de SuperComputación de Cataluña:** 128 procesadores Intel Itanium, 224 cores Intel Xeon, 132 cores AMD Opteron 275. Load Shared Facility.
- **Centro de SuperComputación de Galicia:** 142 nodos de 16 cores Intel Itanium 2 (2272 cores), 1 nodo de 128 cores y 1 nodo de 128 cores. Sun Grid Engine.
- **Centro de SuperComputación y Visualización de Madrid:** 1204 nodos de IBM PowerPC (2744 cores). SLRUM+MOAB.

Gestor de Recursos Teraescale Open-Source Resource and QUEue Manager (Torque)

Diagrama de los Componentes de Torque y funcionalidades

pbs
res
cor
fun
los
rec
el t
caí

pbs_mom: E
trabajo en
informalme
madre de t
ejecución.
ejecución c
trabajo des
una nueva
posible a la
trabajo, ade
devolver al u
trabajo.

pbs_sched: Es el proceso que implementa las políticas de uso definidas, controlando cuando cada trabajo se ejecuta y en que recursos. El Planificador se comunica con los MOMs para interrogarles sobre el estado de los recursos del sistema y con el Servidor para conocer la existencia de trabajos a ejecutar.

JOB

Instalación de Torque

1. Descargar Torque desde http://clusterresources.com/downloads/torque
2. Extraer el paquete descargado:

```
$ tar zxvf torque-2.4.5.tar.gz
$ cd torque-2.4.5
```

3. Configurar (*uso de scp para transferencia de archivos*) y compilar:

```
$ script make-torque-<fecha>.txt
$ ./configure –with-rcp=scp
$ make                          # Unos 10 minutos
$ exit
```

4. Instalar:

```
$ su -          # Requiere instalación como root
$ make install
```

Instalación de Torque

5. Crear paquetes para los nodos de computación:

```
$ make packages
```

6. Instalar el paquete MOM en los nodos de computación:

```
$ ssh root@ccc-nodo1 /apps/sources/torque-2.4.5/
torque-package-mom-linux-i686.sh --install
```

7. Configurar el sistema para que pbs_server, pbs_sched sean servicios:

```
$ cp contrib/init.d/pbs_server /etc/init.d
$ cp contrib/init.d/pbs_sched /etc/init.d
$ chkconfig --add pbs_server
$ chkconfig --add pbs_sched
```

8. Establecer pbs_mom en los clientes como servicio:

```
$ scp contrib/init.d/pbs_mod root@ccc-nodo1:/etc/init.d
$ ssh root@ccc-nodo1 chkconfig --add pbs_mom
$ ssh root@ccc-nodo1 chkconfig --list pbs_mom
```

Configuración Básica de Torque

1. Ejecutar script de inicialización de Torque:

```
$ ./torque.setup root
```

2. Especificar los nodos de computación (/var/spool/torque/server_priv/nodes):

```
$ qmgr –c 'create node ccc-nodo1'
$ qmgr –c 'set node ccc-nodo1 np=4'
```

3. Comprobar que son vistos por el gestor de recursos:

```
$ pbsnodes –l
```

4. Configuración de Torque en los nodos de computación y arranque:

```
$ cat /var/spool/torque/server_name
$ cat /var/spool/torque/pbs_environment
$ service pbs_mom start
```

Configuración Básica de Torque

5. Comprobar que se ven los nodos de computo desde el servidor:

```
$ pbsnodes -a
```

6. Añadir usuarios con privilegios de administración:

```
$ qmgr

Qmgr: set server managers += ccc@ccc-server1
Qmgr: quit
```

7. Comprobar el funcionamiento de la configuración básica:

```
$ qstat –q                  # Verificar conf. colas
$ qmgr –c ‘p s’             # Verificar conf. servidor
$ su – inv1                 # NO SE PUEDEN LANZAR TRABAJOS COMO ROOT
$ echo “sleep 30” | qsub
$ qstat                     # Ver estado del trabajo
```

Configuración Básica de Torque

8. En este punto el trabajo se encontrará en el estado **Q** y no se ejecutará:

```
Job id          Name             User    Time Use  S Queue
--------------- ---------------- ------- --------  - -----
0.ccc-server    STDIN            inv1    0          Q batch
```

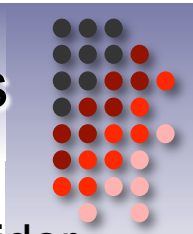
9. Hace falta un planificador (*scheduler*). Se puede usar el que viene con Torque (`pbs_sched`) u otro más avanzado como MAUI:

```
$ service pbs_sched start
```

10. Cambiar el tiempo en el que un trabajo se mantiene en el estado **C**:

```
$ qmgr –c 'set server keep_completed=10'
```

Configuración Básica de Torque: Ajuste de parámetros



11. **scheduler_iteration:** El tiempo, en segundos, entre intentos del servidor para planificar trabajos. Valor por defecto 10 minutos. Recomendable bajarlo a 5 minutos para un cluster pequeño:

```
$ qmgr –c ‘set server scheduler_iteration=300’
```

12. **node_check_rate:** Tiempo, en segundos, que un nodo puede estar sin responder antes de considerarlo no disponible (**down**). Valor por defecto 10 minutos:

```
$ qmgr –c ‘set server node_check_rate=150’
```

13. **node_ping_rate:** Indica el intervalo máximo, en segundos, entre sucesivos ping enviados desde el pbs_server al pbs_mom para determinar el estado del nodo. Valor por defecto 5 minutos:

```
$ qmgr –c ‘s s node_ping_rate=150’
```

14. **tcp_timeout:** Especifica el *timeou*t, en segundos, en el socket TCP entre el pbs_server y el pbs_mom. Valor por defecto 6 segundos.

Configuración Básica de Torque: Ajuste de parámetros del sistema

1. **Aumentar el número de descriptores de archivos:**

```
$ ulimit -n
$ vi /etc/init.d/pbs_mom
ulimit –n 32768
```

2. **Límites en las sesiones interactivas:** 2 métodos

```
$ cat /etc/security/limit.conf
```

```
$ cat /etc/profile
# 256 Mbytes memoria física
# 128 Mbytes segmento de datos
# 15 minutos uso CPU
if [ ! “$PBS_ENVIRONMENT” ] ; then
  if [ ! `whoami` != root ] ; then
     ulimit –m 256000
     ulimit –d 128000
     ulimit –t 900
  fi
fi
```

Envío de trabajos a Torque

- Ejemplo de script de definición de trabajo para Torque:

```
#!/bin/bash
#PBS –N ejemplo1
#PBS –l nodes=1,walltime=00:01:00
#PBS –q batch
#PBS –m ae

date
sleep 10
date
```

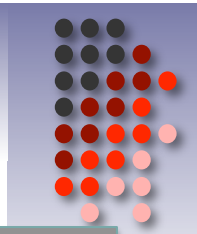
- Envío al sistema de colas y monitorización:

```
$ qsub mitrabajo.pbs
$ qstat
```

- Por omisión sólo root y el usuario que mandó el trabajo y root podrá verlo:

```
$ qmgr –c ‘s s query_other_jobs=true’
```

Envío de trabajos a Torque



```
#!/bin/bash
#PBS -lnodes=1:ppn=1,vmem=5GB
#PBS -m e
#PBS -M luispegg@gmail.com

module load gaussian
export GAUSS_SCRDIR=${GAUSS_SCRDIR}/$PBS_JOBID
mkdir $GAUSS_SCRDIR
# Escribir todo, hasta los chk que queremos recuperar
# en scratch local
cd $GAUSS_SCRDIR
# El archivo de entrada lo tenemos en el mismo directorio desde el
# que mandamos el trabajo y queremos que el log vaya tambien alli
g03 < $PBS_O_WORKDIR/test594.com > $PBS_O_WORKDIR/test594.log
# Si hay algun archivo de checkpoint lo copiamos
cp *.chk $PBS_O_WORKDIR
# Eliminamos el resto de archivos generados y que no hacen falta
# asi como el directorio de trabajo

rm -fr $GAUSS_SCRDIR
```

Envío de trabajos a Torque

- Si el equipo desde el que se envía trabajos es distinto al del servidor:

```
$ qmgr –c ‘set server submit_hosts=ccc-login’
```

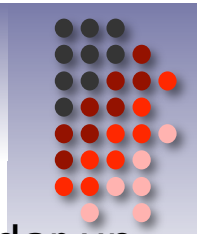
- Establecer el máximo número de trabajos que un usuario puede tener en el sistema:

```
$ qmgr –c ‘set server max_user_run=4’
```

- Definir cola para trabajos sólo secuenciales:

```
$ qmgr –c ‘create queue secuen’
$ qmgr –c ‘s q secuen resources_max.nodect = 1’
$ qmgr –c ‘s q secuen queue_type = Execution’
$ qmgr –c ‘s q secuen max_user_run = 5’
$ qmgr –c ‘s q secuen max_user_queuable = 10’
$ qmgr –c ‘s q secuen enabled = true’
$ qmgr –c ‘s q secuen started = true’
$
```

Envío de trabajos a Torque



- Envío de múltiples trabajos (*JOBS ARRAYS*): Cuando se desea mandar un número elevado de trabajos y todos están basados en el mismo script de definición de trabajo:

```
$ qsub –t 0-3 job_array.pbs
1024.ccc-server
$qstat
1024.0 …
1024.1 …
1024.2 …
1025.3 …
```

- Cada trabajo dentro del ARRAY tiene una variable de entorno (PBS_ARRAYID) establecida a su índice dentro del grupo de trabajos. Esto permitirá, por ejemplo, configurar diferentes acciones para cada trabajo.

```
$ qsub –t 0,10,20,30,40 job_array.pbs
$ qsub –t 0-50,60,70,80 job_array.pbs
```

Gestión de recursos en Torque

- Se pueden definir los siguientes recursos:

```
resources_max.cput=00:30:00
resources_max.walltime=01:00:00
resources_max.mem=512mb
resources_max.vmem=1gb
resources_min.nodect=4
resources_max.nodect=8
resources_max.file=1tb
```

- Se pueden definir colas que sólo puedan usar ciertos usuarios y asignadas sólo a los nodos especificados:

```
# qmgr –c ‘create queue proyecto’
# qmgr –c ‘s q proyecto acl_user_enable = true’
# qmgr –c ‘s q proyecto acl_users = inv1’
# qmgr –c ‘s q proyecto acl_users += inv2’

# qmgr –c ‘create queue server’
# qmgr –c ‘s q server resources_default.neednodes=quad’
# qmgr –c ‘s node ccc-server properties=quad’
```

Copía de seguridad, transferencia de archivos y nodos

- Copia de seguridad de la configuración de Torque y restauración:

```
# qmgr –c 'p s' > /backup/torque/torque_conf.txt
# qmgr < /backup/torque/torque_conf.txt
```

- Indicar como se transfieren los archivos de la salida del trabajo entre los nodos de cómputo y el nodo de envío del trabajo:

```
# vi /var/spool/torque/mom_priv/config

$usecp *:/home_nfs     /home_nfs
```

- Gestión de los nodos de cómputo con pbsnodes:

```
# pbsnodes –l
# pbsnodes –a
# pbsnodes –o ccc-nodo1    # Marcar nodo OFFLINE
# pbsnodes –c ccc-nodo1    # Volver a marcar nodo ONLINE
```

Instalación de MAUI Scheduler

1. Descargar MAUI desde http://clusterresources.com/downloads/maui
2. Extraer el paquete descargado:

```
$ tar zxvf maui-3.3.tar.gz
$ cd maui-3.3
```

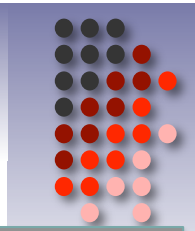
3. Editar msched-common.h si se trata de un cluster grande (>5120 nodos). Configurar y compilar (*por omisión se instala en /usr/local/maui*):

```
$ script make-maui-<fecha>.txt
$ ./configure
$ make                          # Unos 3 minutos
$ exit
```

4. Instalar:

```
$ su -          # Requiere instalación como root
# make install
```

Configuración Básica de MAUI



1. Modificar aquellos valores establecidos por defecto:

```
# vi /usr/local/maui/maui.cfg

# FQDN del servidor
SERVERHOST ccc-server
# Usuario que ejectua el servicio. Mismo que en Torque
ADMIN1      root
SERVERPORT  42559           # Puerto en el que escucha MAUI
RMCFG[CCC-SERVER] TYPE=PBS # Con cual Gestor va a trabajar
SERVERMODE  NORMAL         # NORMAL, TEST, SIMULATION
RMPOLLINTERVAL 00:00:30     # Cada 30 seg. comu. con Torque
```

2. Parar el servicio pbs_sched (además conf. para que no se inicie por defecto):

```
# service pbs_sched stop
# chkconfig pbs_sched off
# vi $M_SRC/contrib/service-scripts/redhat.maui.d
# cp $M_SRC/contrib/service-scripts/redhat.maui.d /etc/init.d/maui
# chkconfig --add maui ; chkconfig maui on
# service maui start
```

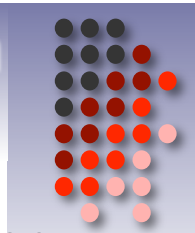
Configuración Básica de MAUI: Ajuste de parámetros

3. **BACKFILLPOLICY:** Permitir que el planificador haga mejor uso de los recursos disponibles cambiando el orden en el que se ejecuta los trabajos. FIRSTFIT, BESTFIT, GREEDY, NONE.
4. **RESERVATIONPOLICY:** Evitar que trabajos largos se queden indefinidamente sin ejecutarse. CURRENTHIGHEST, HIGHEST, NEVER
5. **QUEUETIMEWEIGHT:** Factor por el que se multiplica el tiempo (en minutos) en el que un trabajo está en cola para calcular su prioridad.
6. **JOBNODEMATCHPOLICY**: Indica como deben ser seleccionados los nodos. EXACTNODE -> seleccionar el número de nodos solicitados aunque se pueda empaquetar la tarea en el mismo nodo. (-lnodes=4:ppn=2)
7. **NODEALLOCATIONPOLICY:** MINRESOURCE, MAXBALANCE (cpu speed), FASTEST, PRIORITY, CPULOAD, FIRSTAVAILABLE, PRIORITY

```
# vi maui.cfg

NODEALLOCATIONPOLICY PRIORITY
NODECFG[DEFAULT] PRIORITYF='SPEED + .01 * AMEM - 10 * JOBCOUNT'
```

Configuración Básica de MAUI: Políticas de Utilización



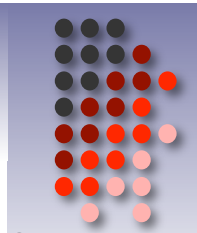
Permiten controlar el flujo de los trabajos en el sistema. Los límites pueden aplicarse a través del conjunto de parámetros ***CFG** (USERCFG, GROUPCFG, CLASSCFG):

- **MAXJOB:** Número máximo de trabajos que pueden estar ejecutándose en el sistema.
- **MAXPROC:** Limita el número de procesadores (cores) que pueden estar ocupándose en cualquier momento.
- **MAXMEM:** Limita la cantidad de memoria (Mbytes) que puede ser utilizado por un trabajo.

```
# vi maui.cfg

USERCFG[DEFAULT]    MAXJOB=4
USERCFG[jlgg]       MAXJOB=8
-------------------------
USERCFG[jlgg]       MAXJOB=2  MAXPROC=24
GROUPCFG[ccc]       MAXJOB=5
CLASSCFG[DEFAULT]   MAXPROC=16
CLASSCFG[batch]     MAXPROC=32
```

Configuración Básica de MAUI: Calidad de Servicio QoS



Permiten dar un tratamiento especial a trabajos, usuarios, grupos y clases:

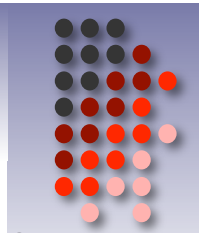
- Asignar una priorización especial. **PRIORITY**
- Asignación de servicios especiales: **PREEMPTOR**, **PREEMPTEE**, **DEDICATED**.
- Ignorar límites establecidos por omisión o establecer otros nuevos: **IGNMAXJOB**, **MAXJOB**.

```
# vi maui.cfg

QOSCFG[hi]  PRIORITY=100 FLAGS=PREEMPTOR IGNMAXJOB
QOSCFG[low] PRIORITY=-1000 FLAGS=PREEMPTEE
------------------------
GROUPCFG[ccc]        QLIST=hi:low QDEF=hi
CLASSCFG[interact]   FLAGS=PREEMPTOR
CLASSCFG[batch]      FLAGS=PREEMPTEE
```

- qsub –lnodes=4,qos=hi myboj.cmd

Configuración Básica de MAUI: Calidad de Servicio QoS



Permiten dar un tratamiento especial a trabajos, usuarios, grupos y clases:

- Asignar una priorización especial. **PRIORITY**
- Asignación de servicios especiales: **PREEMPTOR**, **PREEMPTEE**, **DEDICATED**.
- Ignorar límites establecidos por omisión o establecer otros nuevos: **IGNMAXJOB**, **MAXJOB**.

```
# vi maui.cfg

QOSCFG[hi]  PRIORITY=100 FLAGS=PREEMPTOR IGNMAXJOB
QOSCFG[low] PRIORITY=-1000 FLAGS=PREEMPTEE
------------------------
GROUPCFG[ccc]        QLIST=hi:low QDEF=hi
CLASSCFG[interact]   FLAGS=PREEMPTOR
CLASSCFG[batch]      FLAGS=PREEMPTEE
```

- qsub –lnodes=4,qos=hi myboj.cmd

Configuración Básica de MAUI: Reservas

Mecanismo con el que se garantiza la disponibiliad de unos determinados recursos durante un tiempo determinado:

- En el ejemplo se define una Reserva desde las 8 de la mañana a las 5 de la tarde en 20 nodos del sistema con un número de tareas (procesadores) de 40, los Lunes, Martes y Miércoles.

```
# vi maui.cfg
# Define la reserva fast

SRCFG[fast] STARTTIME=08:00:00 ENDTIME=17:00:00
SRCFG[fast] HOSTLIST=nodo0[1-20]
SRCFG[fast] TASKCOUNT=40
SRCFG[fast] DAYS=MON TUE WED
SRCFG[fast] TIMELIMIT=01:30:00
```

- qsub –lnodes=10:ppn=2,walltime=1:00:00 -W x=FLAGS:ADVRES:fast testjob.cmd