

Envío y gestión de los trabajos

Enviar los trabajos al sistema de colas

El envío de trabajos se realiza a través del comando `qsub`, cuyo argumento obligatorio es el nombre de un script de shell.

```
ct$ qsub script.sh
```

El comando `qsub` admite como parámetros las mismas opciones que pueden indicarse como comentarios `#PBS` en el script.

Consultar el estado del trabajo, las colas o los nodos

Información de las colas

El comando `qstat` permite consultar el estado de las colas,

```
ct$ qstat -q # Información global de las colas
server: ctcomp2
```

Queue	Memory	CPU	Time	Walltime	Node	Run	Que	Lm	State
graphic32	--	--		04:00:00	--	0	0	--	E R
np16	--	--		192:00:0	--	0	0	--	E R
np32	--	--		288:00:0	--	0	0	--	E R
especial	--	--		672:00:0	--	0	0	--	E R
parallel	--	--		192:00:0	--	0	0	--	E R
np2	--	--		192:00:0	--	0	0	--	E R
np8	--	--		192:00:0	--	0	0	--	E R
short	--	--		--	--	0	0	--	E R
graphic1	--	--		04:00:00	--	0	0	--	E R
np1	--	--		672:00:0	--	0	0	--	E R
batch	--	--		--	--	0	0	--	E R
np4	--	--		192:00:0	--	0	0	--	E R
interactive	--	--		01:00:00	--	0	0	--	E R
np64	--	--		384:00:0	--	0	0	--	E R
graphic	--	--		--	--	0	0	--	E R
bigmem	--	--		--	--	0	0	--	E R
graphic8	--	--		04:00:00	--	0	0	--	E R
						0	0		

La columna State indica con su primera letra si la cola está (E)nabled o (D)isabled y con la segunda letra si la cola está (R)unning o (S)topped.

Información de los trabajos

Cada vez que se envía un trabajo se le asigna un JOB_ID que sirve como identificador único. Si el trabajo se envió con la opción -t entonces se identificará mediante job_id[indice].

```
ct$ qstat # Información general de los trabajos de usuario
Job id          Name          User          Time Use S
Queue
-----
--
999999.ctcomp2 nombre_del_trabajo nombre_usuario 38:05:59 R
np32
```

La columna Time Use muestra el tiempo de CPU usado. La columna S es el estado del trabajo, que puede ser uno de los siguientes:

- C - Job is completed after having run
- E - Job is exiting after having run.
- H - Job is held.
- Q - job is queued, eligible to run or routed.
- R - job is running.
- T - job is being moved to new location.
- W - job is waiting for its execution time(-a option) to be reached.

```
ct$ qstat -f 999999.ctcomp2 # Información sobre un trabajo específico
Job Id: 999999.ctcomp2.innet
Job_Name = nombre_del_trabajo
Job_Owner = nombre_usuario@ctcomp2.innet
job_state = Q
queue = np32
server = ctcomp2.innet
Checkpoint = u
ctime = Fri Feb 12 10:09:34 2016
Error_Path =
ctcomp2.innet:/home/local/nombre_usuario/nombre_del_trabajo.e999999
Hold_Types = n
Join_Path = n
Keep_Files = n
Mail_Points = ae
Mail_Users = nombre_usuario@usc.es
mtime = Fri Feb 12 10:09:34 2016
Output_Path =
ctcomp2.innet:/home/local/nombre_usuario/nombre_trabajo.o999999
Priority = 0
qtime = Fri Feb 12 10:09:34 2016
Rerunable = True
```

```
Resource_List.neechnodes = 1:ppn=32:intel:xeonl
Resource_List.nodect = 1
Resource_List.nodes = 1:ppn=32:intel:xeonl
Resource_List.vmem = 63gb
Resource_List.walltime = 12:00:00
substate = 10
Variable_List = PBS_0_QUEUE=batch,PBS_0_HOME=/home/local/nombre_usuario,
PBS_0_LOGNAME=nombre_usuario,
PBS_0_PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games,
PBS_0_MAIL=/var/mail/nombre_usuario,PBS_0_SHELL=/bin/bash,
PBS_0_LANG=es_ES.UTF-8,PBS_0_WORKDIR=/home/local/nombre_usuario,
PBS_0_HOST=ctcomp2.innet,PBS_0_SERVER=ctcomp2
euser = nombre_usuario
egroup = citius
queue_rank = 2110
queue_type = E
etime = Fri Feb 12 10:09:34 2016
submit_args = script.sh
fault_tolerant = False
job_radix = 0
submit_host = ctcomp2.innet
```

Una característica interesante de los trabajos finalizados es el EXIT_STATUS que se mostraría cuando el JOB_STATE es C.

Codigo interno	Valor de EXIT_STATUS	Significado
JOB_EXEC_OVERLIMIT	-10	
JOB_EXEC_STDOUTFAIL	-9	
JOB_EXEC_CMDFAIL	-8	Exec() of user command failed
JOB_EXEC_BADRESRT	-7	Job restart failed
JOB_EXEC_INITRMG	-6	Job aborted on MOM init, chkpt, ok migrate
JOB_EXEC_INITRST	-5	Job aborted on MOM init, chkpt, no migrate
JOB_EXEC_INITABT	-4	Job aborted on MOM initialization
JOB_EXEC_RETRY	-3	Job execution failed, do retry
JOB_EXEC_FAIL2	-2	Job execution failed, after files, no retry
JOB_EXEC_FAIL1	-1	Job execution failed, before files, no retry
JOB_EXEC_OK	0	Job execution successful
	1-256	Exit status of the top-level shell
	>256	Trabajo terminado por una señal UNIX, restarle 256 nos da el número de la señal.

```
ct$ checkjob 999999.ctcomp2 # Información sobre un trabajo específico
checking job 999999

State: Running
Creds: user:nombre_usuario group:citius class:np32 qos:DEFAULT
WallTime: 00:25:46 of 12:00:00
```

```
SubmitTime: Tue Feb 16 10:40:31  
(Time Queued Total: 00:00:01 Eligible: 00:00:01)
```

```
StartTime: Tue Feb 16 10:40:32  
Total Tasks: 32
```

```
Req[0] TaskCount: 32 Partition: DEFAULT  
Network: [NONE] Memory >= 0 Disk >= 0 Swap >= 0  
Opsys: [NONE] Arch: [NONE] Features: [active][intel][xeonl]  
Allocated Nodes:  
[inode15:32]
```

```
IWD: [NONE] Executable: [NONE]  
Bypass: 0 StartCount: 1  
PartitionMask: [ALL]  
Flags: RESTARTABLE
```

```
Reservation '137092' (-00:25:32 -> 11:34:28 Duration: 12:00:00)  
PE: 32.00 StartPriority: 21
```

```
ct$ tracejob -n 3 999999.ctcomp2 # Devuelve el contenido de los logs  
relativos al jobid indicado.  
Job: 136553.ctcomp2.innet
```

```
02/10/2016 15:22:26 S enqueueing into batch, state 1 hop 1  
02/10/2016 15:22:26 S dequeuing from batch, state QUEUED  
02/10/2016 15:22:26 S enqueueing into np1, state 1 hop 1  
02/10/2016 15:22:26 S Job Run at request of citiuscap@ctcomp2.innet  
02/10/2016 15:22:26 S Not sending email: User does not want mail of this  
type.  
02/10/2016 15:22:26 A queue=batch  
02/10/2016 15:22:26 A queue=np1  
02/10/2016 15:22:26 A user=nombre_usuario group=citius  
jobname=nombre_trabajo queue=np1 ctime=1455114146  
qtime=1455114146 etime=1455114146 start=1455114146  
owner=nombre_usuario@ctcomp2.innet exec_host=inode19/24  
Resource_List.nodect=1  
Resource_List.nodes=1:ppn=1 Resource_List.vmem=2040mb  
Resource_List.walltime=12:00:00  
02/10/2016 16:08:34 S Exit_status=0 resources_used.cput=00:46:14  
resources_used.mem=234868kb  
resources_used.vmem=1002480kb  
resources_used.walltime=00:46:08  
02/10/2016 16:08:34 S on_job_exit valid pjob: 999999.ctcomp2.innet  
(substate=50)  
02/10/2016 16:08:34 A user=nombre_usuario group=citius  
jobname=nombre_trabajo queue=np1 ctime=1455114146 qtime=1455114146  
etime=1455114146 start=1455114146 owner=nombre_usuario@ctcomp2.innet  
exec_host=inode19/24 Resource_List.nodect=1
```

```
Resource_List.nodes=1:ppn=1 Resource_List.vmem=2040mb
Resource_List.walltime=12:00:00 session=7304 end=1455116914 Exit_status=0
resources_used.cput=00:46:14 resources_used.mem=234868kb
resources_used.vmem=1002480kb resources_used.walltime=00:46:08
02/10/2016 17:08:35 S dequeuing from np1, state COMPLETE
```

Información de los nodos

Para obtener una vista global del estado del cluster se puede utilizar el comando nodes - usage.

```
$ nodes - usage
+-----+-----+
| USAGE                                | NODE                                |
+-----+-----+
| #####                               | node1 (64/64)                       |
| #####                               | node2 (64/64)                       |
| #####                               | node3 (0/64)                        |
| #####                               | node4 (64/64)                       |
| #####                               | node5 (0/64)                        |
| #####                               | node6 (64/64)                       |
| #####                               | node7 (0/64)                        |
| #####                               | inode11 (0/32)                      |
| #####                               | inode12 (0/??)                      |
| #####                               | inode13 (0/32)                      |
| #####                               | inode14 (0/32)                      |
| #####                               | inode15 (0/??)                      |
| #####                               | inode16 (0/32)                      |
| #####                               | inode17 (0/??)                      |
| #####                               | inode18 (0/??)                      |
| ##                                   | inode19 (2/32)                      |
| #####                               | inode20 (28/32)                     |
+-----+-----+
| #####                               | TOTAL (286/640)                     |
+-----+-----+
```

Para obtener información sobre los usuarios que se encuentran en cada nodo, se puede utilizar el comando node-users <nodo>:

```
$ node-users node1
Tracing node
jobs.....
jorge.suarez natalia.fernandez
```

Para obtener información más detallada sobre los nodos, se puede utilizar el comando pnbsnodes:

```
ct$ pnbsnodes #Información detallada de todos los nodos
node1
state = free
np = 64
```

```
properties = amd,bigmem,test,active,activeX
ntype = cluster
status =
rectime=1455267717,varattr=,jobs=,state=free,netload=86957182662,gres=,loadave=0.00,ncpus=64,physmem=132250896kb,availmem=162914704kb,totmem=163499276kb,
, idletime=1876325,nusers=0,nsessions=0,uname=Linux node1 3.2.0-4-amd64 #1
SMP Debian 3.2.51-1 x86_64,opsys=linux
mom_service_port = 15002
mom_manager_port = 15003
gpus = 0

node2
state = down,offline
np = 64
properties = amd,bigmem
ntype = cluster
status =
rectime=1454919087,varattr=,jobs=,state=free,netload=1185896,gres=,loadave=0.00,ncpus=64,physmem=264633540kb,availmem=295220244kb,totmem=295881920kb, idl
etime=11140,nusers=0,nsessions=0,uname=Linux node2 3.2.0-4-amd64 #1 SMP
Debian 3.2.51-1 x86_64,opsys=linux
mom_service_port = 15002
mom_manager_port = 15003
gpus = 0

.....
```

```
ct$ pbsnodes -l # Listado de los nodos apagados(down) o no
disponibles(offline)
node2          down,offline
node3          down,offline
node4          down,offline
node5          down,offline
node6          down,offline
node7          down,offline
inode11        down,offline
inode12        down,offline
inode13        down,offline
inode14        down,offline
inode15        down,offline
inode17        down,offline
inode18        down,offline
inode19        down,offline
```

Eliminar un trabajo de la cola

El comando `qdel` permite al usuario eliminar un trabajo. Funciona enviándole primero una señal

TERM y luego una KILL. Este comando necesita como argumento el identificador que PBS le asigna cuando se registra un nuevo trabajo, y que se puede consultar con qstat.

```
ct$ qdel job_id
```

From:

<https://wiki.citius.usc.es/> - **Wiki do CiTIUS**

Permanent link:

https://wiki.citius.usc.es/centro:servizos:hpc:envio_trabajo?rev=1497615716

Last update: **2017/06/16 14:21**

