

## Monitorización

El sistema de colas tiene básicamente dos modos de monitorizar los trabajos enviados:

- Mediante los ficheros de salida generados por el job.
- Mediante comandos directamente en la ventana de ssh.

En esta parte definiremos los ficheros de salida y los principales comandos utilizados junto con la salida por pantalla que generan dichos comandos.

Al ejecutar `enviamatlab` (cualquiera de las versiones) se generan de forma automática dos ficheros en el path que contiene el script ejecutado (p.e: `script.m`) cuya nomenclatura viene dada por:

- **matlabqsub.20091123T105716**: este fichero lo genera el `enviamatlab` y será el que arranque matlab y ejecute el `script.m`. El final del nombre es la fecha de ejecución.
- **script.20091123T105716.log**: en este fichero se recoge la salida por pantalla de Matlab y `script` representa el nombre de archivo `.m` ejecutado. De nuevo es la fecha la que se muestra tras el nombre del script.

Una vez el trabajo ha finalizado se genera un nuevo fichero **script.o114037**, donde el número indica el identificador del job ejecutado, en el que se recoge la salida completa del job. Si se ha incorporado una dirección de correo electrónico, también se recibirá un e-mail indicando el final del job con el asunto `PBS JOB JobId.ce01.macc.unican.es`. En alguna versión del `enviamatlab` también se genera un fichero de error llamado **script.e114037** de forma equivalente a la estructura del fichero de salida.

Desde `oceano` podemos utilizar distintos comandos u órdenes para consultar el estado de un trabajo, de la cola, etc. Aunque normalmente habría que conectarse a la máquina `ce01` anteponiendo el comando `ssh ce01` a la orden que queramos ejecutar (`qstat`, `qdel`, etc.), este comando puede omitirse ya que la propia máquina *autocompleta* la orden. Introduzcamos los más útiles:

**qstat**: Nos muestra por pantalla una tabla en la podemos observar el estado de la cola de trabajos. En caso de que no devuelva nada se entiende que no existe ningún trabajo activo ni en espera. La sintaxis es la siguiente: `qstat`. Si queremos que devuelva una información más detallada podemos incluir *flags* como `-n` o `-nl`.

```
[rodri@oceano ejemplo]$ qstat -nl
ce01.macc.unican.es:
Job ID          Username Queue   Jobname      SessID NDS   Req'd Req'd  Elap
-----
120765.ce01.macc.uni lluis   hmem    Psw0000__   22862   1  --   --   -- R 06:06
wn023+wn023+wn023+wn023+wn023+wn023+wn023+wn023
120770.ce01.macc.uni lluis   hmem    Psw0000__   2766    1  --   --   -- R 06:06
wn018+wn018+wn018+wn018+wn018+wn018+wn018+wn018
120772.ce01.macc.uni lluis   hmem    Psw0000__   6314    1  --   --   -- R 06:06
wn015+wn015+wn015+wn015+wn015+wn015+wn015+wn015
120773.ce01.macc.uni lluis   hmem    Psw0000__  14094    1  --   --   -- R 06:06
wn014+wn014+wn014+wn014+wn014+wn014+wn014+wn014
...
```

Al igual que para otras órdenes de unix, el comando `grep` limita la búsqueda a aquellos elementos que contengan la expresión que se pasa a continuación. Por ejemplo, si sólo queremos que se muestren los trabajos de un usuario concreto utilizaremos `qstat -n | grep nombreUsuario`. De este modo el comando seleccionará sólo las líneas en las cuales aparece el `nombreUsuario`.

```
[rodri@oceano ejemplo]$ qstat -n | grep rodri
121138.ce01.macc.uni rodri    macc    val_1    28628    1  --   --  48:00 R 05:30
121139.ce01.macc.uni rodri    macc    val_2    31232    1  --   --  48:00 R 03:37
121176.ce01.macc.uni rodri    macc    test     29987    1  --   --  48:00 R 00:05
```

```
121177.ce01.macc.uni rodri macc val_3 30127 1 -- -- 48:00 C 00:01
```

Algunos de los parámetros mas representativos:

- **Job ID:** identificador del trabajo (asignado por PBS).
- **Username:** propietario del trabajo (usuario que lo envió).
- **Queue:** cola en la que reside el trabajo.
- **Jobname:** nombre del trabajo dado por el usuario o establecido por PBS en caso de no especificarse.
- **NDS:** número de nodos solicitados.
- **TSK:** número de tareas solicitadas.
- **Req'd Memory:** cantidad de memoria solicitada.
- **Req'd Time:** tiempo de reloj (wallclock) solicitado.
- **S:** estado del trabajo:
  - E el trabajo está saliendo después de finalizar su ejecución.
  - H el trabajo está capturado.
  - Q el trabajo está en cola, elegible para su ejecución.
  - R el trabajo está ejecutándose.
  - T el trabajo está en transición (moviéndose a otra localización).
  - W el trabajo está a la espera de alcanzar el tiempo de ejecución.
  - S el trabajo está suspendido.
- **Elap Time:** tiempo de CPU utilizado.
- En el caso de utilizar el flag **-n1** también nos aparecen los nodos utilizados por el job.

**checkjob:** Muestra por pantalla la información del trabajo que le especifiquemos mediante el JobId. Se ejecuta del siguiente modo:ssh ce01 checkjob JobId

```
[rodri@oceano ejemplo]$ ssh ce01 checkjob 121176
Scientific Linux CERN SLC release 4.6 (Beryllium)
checking job 121176
State: Running
Creds: user:rodri group:gmeteo class:macc qos:DEFAULT
WallTime: 00:07:29 of 2:00:00:00
SubmitTime: Wed Dec 9 18:18:11
(Time Queued Total: 00:00:02 Eligible: 00:00:02)
StartTime: Wed Dec 9 18:18:13
Total Tasks: 1
Req[0] TaskCount: 1 Partition: DEFAULT
Network: [NONE] Memory >= 0 Disk >= 0 Swap >= 0
Opsys: [NONE] Arch: [NONE] Features: [macc]

Allocated Nodes:
[wn007.macc.unican.es:1]

IWD: [NONE] Executable: [NONE]
Bypass: 0 StartCount: 1
PartitionMask: [ALL]
Flags: BACKFILL
Reservation '121176' (-00:07:48 -> 1:23:52:12 Duration: 2:00:00:00)
PE: 1.00 StartPriority: 1
```

**tracejob:** Da información sobre los tiempos de trabajo del trabajo especificado por el JobId. Estos tiempos pueden consultarse tanto antes como después de la finalización del trabajo. La llamada a esta función es:

ssh ce01 tracejob JobId

```
[rodri@oceano ejemplo]$ ssh ce01 tracejob 121176

Scientific Linux CERN SLC release 4.6 (Beryllium)
```

```
/var/spool/pbs/server_priv/accounting/20091209: Permission denied
/var/spool/pbs/mom_logs/20091209: No such file or directory
/var/spool/pbs/sched_logs/20091209: No such file or directory
```

```
Job: 121176.ce01.macc.unican.es
```

```
12/09/2009 18:18:11 S   enqueueing into macc, state 1 hop 1
12/09/2009 18:18:11 S   Job Queued at request of rodri@oceano.macc.unican.es, owner = rodri@oceano.macc.unican.es, job n
12/09/2009 18:18:13 S   Job Modified at request of root@ce01.macc.unican.es
12/09/2009 18:18:13 S   Job Run at request of root@ce01.macc.unican.es
12/09/2009 18:18:13 S   Job Modified at request of root@ce01.macc.unican.es
```

**qdel:** Elimina un trabajo de la cola antes de que finalice. Para ello escribimos:

```
ssh ce01 qdel JobId
```

```
[rodri@oceano ejemplo]$ qdel 121176
```

**qsig:** Este comando lo usaremos cuando se nos quede colgado un trabajo y no responda al comando qdel. Su sintaxis es la siguiente:

```
ssh ce01 qsig -s TERM JobId
```

```
[rodri@oceano ejemplo]$ ssh ce01 qsig -s TERM 121176
```

#### Otros comandos:

- **qalter:** Modificar los atributos de un trabajo encolado y pendiente de ejecución.
- **qhold:** Marcar un trabajo como no elegible para su ejecución.
- **qlrs:** Liberar un trabajo del estado de espera establecido con el comando qhold.
- **qorder:** Cambiar el orden de dos trabajos en espera de ser ejecutados.
- **qmove:** Mover trabajos de una cola a otra.
- **man pbs\_resources:** Información sobre los requerimientos que podemos usar.
- **more** y **cat:** Estos comandos nos permiten visualizar por pantalla archivos de texto. Con ellos podemos visualizar los archivos de salida del job mediante: `more/cat nombreArchivo`.