
Envío de trabajos Matlab al cluster

Dentro de las máquinas del cluster hay una que se encarga de distribuir los trabajos, asignando nodos a los jobs según estén o no disponibles para ser utilizados por la cola especificada (**macc**, **hmem**, etc.) por el script a ejecutar. Esta máquina es ce01.macc.unican.es (submitting machine). Salvo necesidad expresa la cola común al grupo es la cola **macc** siendo el resto de colas generadas para uso específico ya sea de gente del grupo meteo (modelización, portal, etc.) o de otros grupos de investigación y proyectos (EELA, ENSEMBLES, etc.).

Conexión a oceano

La conexión a oceano la realizaremos mediante SSH. Para ello necesitamos conocer el nombre de la máquina a la cual nos queremos conectar (oceano.macc.unican.es, mar.macc.unican.es), nuestro nombre de usuario y nuestra contraseña. De este modo nos situaremos directamente en nuestro home ('/oceano/gmeteo/users/nombreusuario'). En caso de desconocer el nombre de usuario o la contraseña podéis poneros en contacto con Antonio.

Envío de trabajos

NOTA: La primera (y **sólo la primera**) vez que queramos enviar un trabajo de Matlab al cluster debemos editar el fichero .bashrc de nuestro home de oceano, incluyendo al final del mismo las líneas:

```
export EMAIL=tuemail[at]unican.es [[BR]] source /software/etc/bashrc
```

El fichero final debería ser tal que así (se ha cogido como ejemplo el .bashrc del home de Rodri):

```
# .bashrc

# Source global definitions

if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific aliases and functions

#      alias      pgf90      /software/pgi/linux86-64/7.1-6/bin/pgf90

export SVN_EDITOR=vi

#source /software/ScientificLinux/4.6/etc/bashrc

export EMAIL="rmanzanas@ifca.unican.es"

source /software/etc/bashrc
```

En la actualidad están instalados en el cluster las versiones de Matlab6.5 y Matlab2009, y ambas pueden ser utilizadas para realizar trabajos en la cola mediante órdenes similares. En el primer caso el script a necesario está incluido en el cluster (\software\bin\enviamatlab) y os ejecuta cualquier script de matlab que esté en la carpeta en la que os encontréis mediante la orden:

```
enviamatlab script.m
```

De este modo, se generarán en la carpeta dos ficheros:

- **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.

Al final de la ejecución del script.m se genera un nuevo fichero **script.o114037**, donde el número indica el identificador del job ejecutado.

Para realizar trabajos con Matlab2009 es necesario generar un enviamatlab2009.sh equivalente al que se usa para el Matlab6.5 pero que arranque directamente la última versión de matlab ya que el cluster, por defecto, arranca la anterior. Dicha función podeis pedírsela a Rodri, Sixto o descargarla directamente en esta página, guardarla en vuestro home y exportar el path donde se encuentra mediante la orden:

```
EXPORT PATH="/oceano/gmeteo/users/nombreusuario"
```

Si el script se llama enviamatlab2009.sh, su ejecución es equivalente a la expuesta en el caso anterior sin más que cambiar enviamatlab por enviamatlab2009.sh. De igual modo, se generarán los ficheros anteriores durante la ejecución del job.

Una vez tenemos estas funciones y nos hemos conectado a oceano, para enviar el trabajo debemos situarnos en el directorio en el cual se encuentre el script que queremos ejecutar. Para realizar el envío, las funciones enviamatlab2009.sh o enviamatlab65.sh tienen que estar en el **mismo directorio** que el script en cuestión.

En todos los ejemplos siguientes se ha utilizado enviamatlab2009.sh, pero si se quisiese realizar un trabajo con la versión 6.5 de Matlab bastaría simplemente con sustituir enviamatlab2009.sh por enviamatlab65.sh

Una vez ejecutamos la función nos aparecerá por pantalla una serie de mensajes, como se muestra en el ejemplo, en los que se muestra la información del trabajo enviado. En la última línea se muestra el identificador del trabajo (**JobId**, en el ejemplo **121176**) seguido del nombre de la máquina, ce01.macc.unican.es. Este identificador nos servirá para informarnos del estado del trabajo.

Los mensajes devueltos por pantalla hacen referencia a distintas características del trabajo. Dentro de ellas podemos destacar:

- PBS -q: hace referencia a la cola utilizada (ej: macc)
- PBS -l nodes=!1:ppn=1. Esta orden adquiere relevancia sobretodo en los trabajos realizados con Matlab2009, ya que nos permite especificar cuantos procesador de un nodo queremos utilizar (2, 4, 8 dependiendo de las características de la máquina).
- PBS -l walltime: es el límite temporal del job. Para disponer de más tiempo de ejecución debemos modificar este argumento.
- PBS -M: especifica la dirección de correo electrónico a la cual mandará un aviso cuando acabe el job.

```
[rodri@oceano ejemplo]$ ./enviamatlab2009.sh test
```

```
#!/bin/bash
```

```
#PBS -N test
```

```
#PBS -q macc
```

```
#PBS -l nodes=1:ppn=1
```

```
#PBS -l walltime=48:00:00
```

```
#PBS -m ea
```

```
#PBS -M rmanzanas@ifca.unican.es
```

```
#PBS -r n

export LD_ASSUME_KERNEL=2.6.18

export PATH=/software/Matlab_R2009a/bin:$PATH

matlab -logfile /vols/mar/users/rodri/ejemplo/logs/log_20091209181811.err -r "cd /vols/mar/users/rodri/ejemplo;test;" -nos

exit 0

121176.ce01.macc.unican.es
```

Observaciones

En este apartado podéis escribir las observaciones, dificultades o aclaraciones que encontréis o creáis necesarias a la hora de usar el cluster.

- Como ya se ha indicado, para poder mandar jobs al cluster debemos tener las función `enviamatlab2009.sh` o `enviamatlab65.sh` (quien las necesite se las puede pedir a Rodri) en el mismo directorio que el script que a ejecutar. Una vez copiada/s en el directorio correspondiente y después de habernos asegurado de que está/n correctamente implementada/s, es decir, tiene nuestro nombre de usuario y las demás variables que nos identifican correctamente, podemos ejecutarla/s sin problema alguno siguiendo las instrucciones explicadas anteriormente.
- Es importante tener en cuenta que el Matlab en el que se arranca el proceso no tiene porque tener inicializada MeteoLab y que, por lo tanto, debemos añadir al script una orden para que la inicialice. Bastaría con añadir al inicio del script la ruta de acceso al MeteoLab que utilizemos (es recomendable tenerla siempre actualizada en nuestro home), ejecutar el programa `init` (que arranca la toolbox) y volver al directorio en el que queremos trabajar. De este modo podremos trabajar con todas las funciones de MeteoLab.
- A la hora de marcar las rutas en Linux debemos usar los elementos propios de este sistema. En particular debemos usar la `/` en lugar de la `\` usada habitualmente en Windows. Es importante crear los scripts de modo que puedan ser entendidos en Linux. Para consultar dudas podéis preguntar a Chus, Antonio, etc.
- Las funciones de Meteolab ya están escritas de modo que puedan ejecutarse en Linux y Windows indistintamente (en principio).
- Siempre que tengamos algún problema con la sintaxis de un comando o no sepamos exactamente qué es lo que hace podemos acudir a su manual. Por ejemplo, en el caso del comando `qstat` (monitorización de trabajos) bastaría con teclear `man qstat`.

```
[rodri@oceano ejemplo]$ man qstat

qstat(1B)                                PBS                                qstat(1B)

NAME

    qstat - show status of pbs batch jobs

SYNOPSIS

    qstat [-f [-1]][-W site_specific] [job_identifier... | destination...]

    qstat [-a|-i|-r|-e] [-n [-1]] [-s] [-G|-M] [-R] [-u user_list] [job_identifier... | destination...]
```

```
qstat -Q [-f [-1]][-W site_specific] [destination...]
```

```
qstat -q [-G|-M] [destination...]
```

```
qstat -B [-f [-1]][-W site_specific] [server_name...]
```

DESCRIPTION

The `qstat` command is used to request the status of jobs, queues, or a batch server. The requested status is written to standard out.

When requesting job status, synopsis format 1 or 2, `qstat` will output information about each job identifier or all jobs at each destination. Jobs for which the user does not have status privilege are not displayed.

When requesting queue or server status, synopsis format 3 through 5, `qstat` will output information about each destination.

OPTIONS

`-f` Specifies that a full status display be written to standard out.

: