

Universitat de Lleida

Grado de Informàtica

Escola Politècnica Superior

Computación Distribuida y Aplicaciones

Cores Prado, Fernando

Sun Grid Engine

tutorial SGE

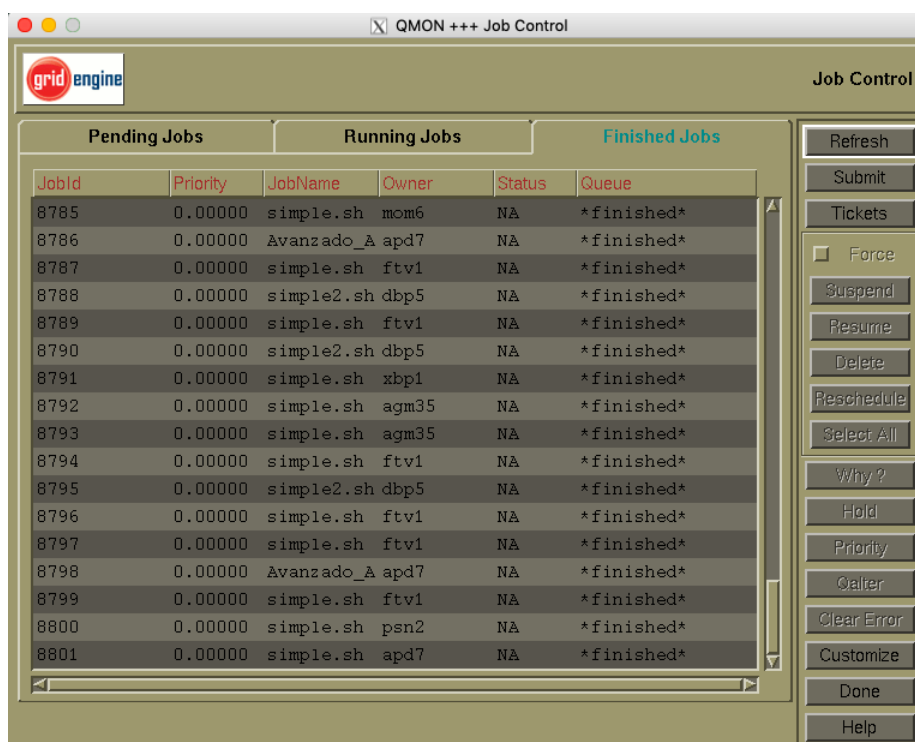
Pérez Datsira, Albert

April 24, 2020

1 Ejercicio 1: Envío de Trabajos

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 89x14
[apd7@moore ~]$ ls
avanzado.sh simple.sh
[apd7@moore ~]$ qsub simple.sh
Your job 8801 ("simple.sh") has been submitted
```

Figure 1: Fichero simple.sh subido y lanzado trabajo para ejecución con id 8801



The image shows a window titled "QMON +++ Job Control" with a "grid engine" logo. It displays a table of jobs with columns: JobId, Priority, JobName, Owner, Status, and Queue. The table is divided into three sections: Pending Jobs, Running Jobs, and Finished Jobs. Job 8801 is listed in the Finished Jobs section with a status of NA.

JobId	Priority	JobName	Owner	Status	Queue
8785	0.00000	simple.sh	mom6	NA	*finished*
8786	0.00000	Avanzado_A	apd7	NA	*finished*
8787	0.00000	simple.sh	ftv1	NA	*finished*
8788	0.00000	simple2.sh	dbp5	NA	*finished*
8789	0.00000	simple.sh	ftv1	NA	*finished*
8790	0.00000	simple2.sh	dbp5	NA	*finished*
8791	0.00000	simple.sh	xbp1	NA	*finished*
8792	0.00000	simple.sh	agm35	NA	*finished*
8793	0.00000	simple.sh	agm35	NA	*finished*
8794	0.00000	simple.sh	ftv1	NA	*finished*
8795	0.00000	simple2.sh	dbp5	NA	*finished*
8796	0.00000	simple.sh	ftv1	NA	*finished*
8797	0.00000	simple.sh	ftv1	NA	*finished*
8798	0.00000	Avanzado_A	apd7	NA	*finished*
8799	0.00000	simple.sh	ftv1	NA	*finished*
8800	0.00000	simple.sh	psn2	NA	*finished*
8801	0.00000	simple.sh	apd7	NA	*finished*

Figure 2: Trabajo 8801 finalizado

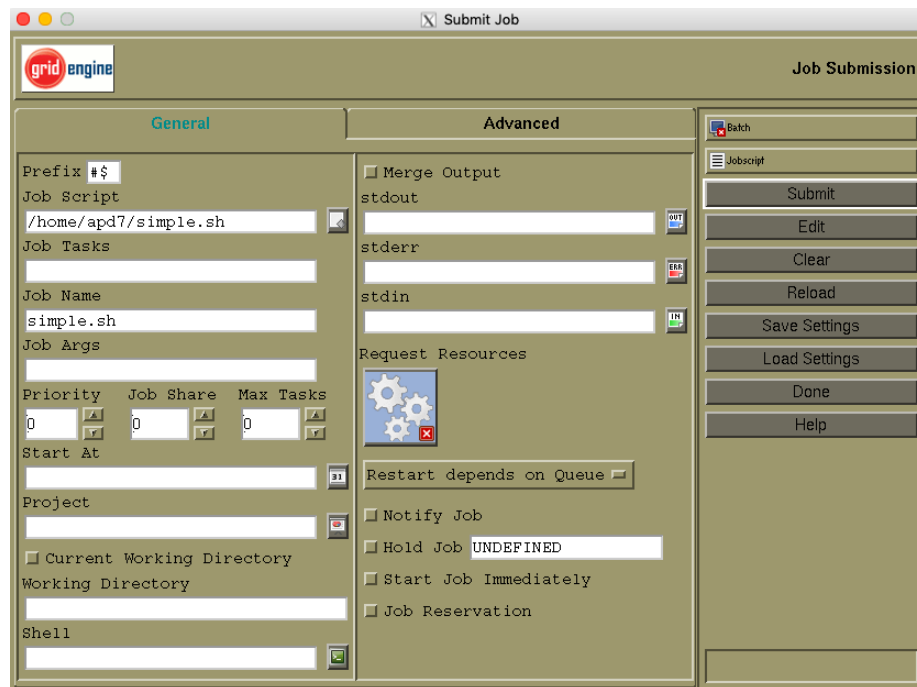


Figure 3: Lanzar trabajo simple.sh con qmon

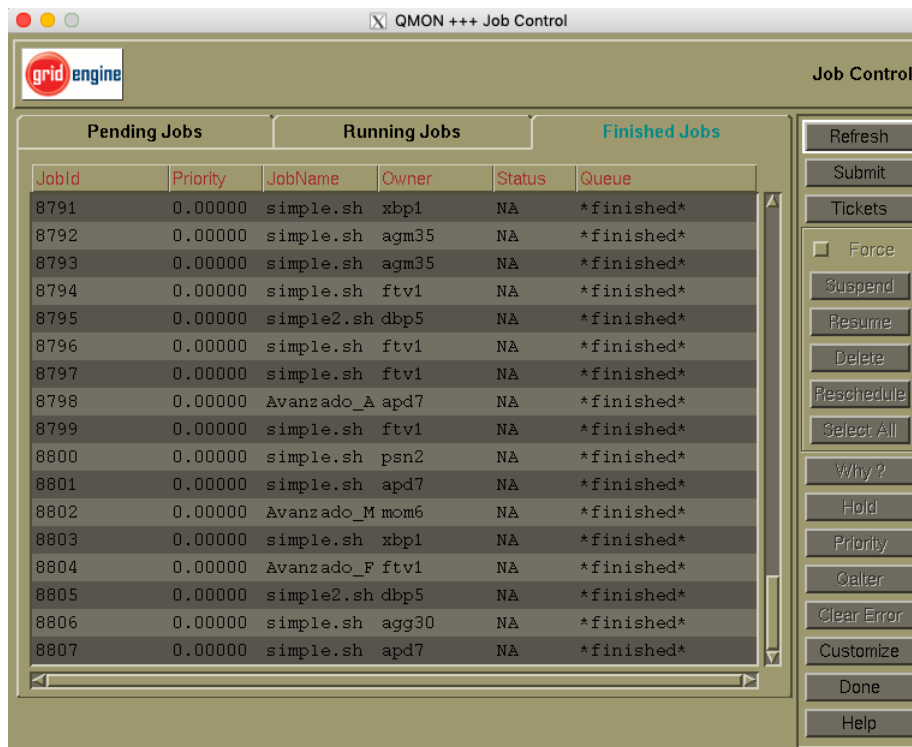


Figure 4: Trabajo 8807 finalizado

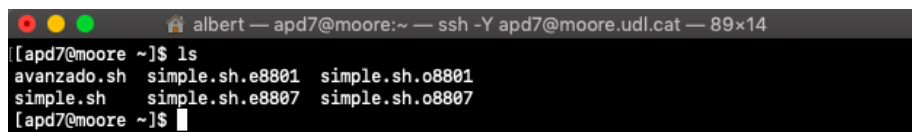


Figure 5: Ficheros de salida

2 Ejercicio 2: Opciones Trabajos

2.1 Ejercicio 2a

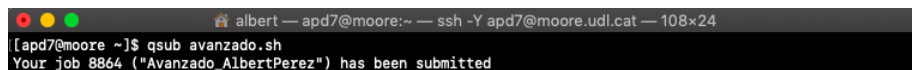


Figure 6: Lanzamos trabajo avanzado.sh con id 8864

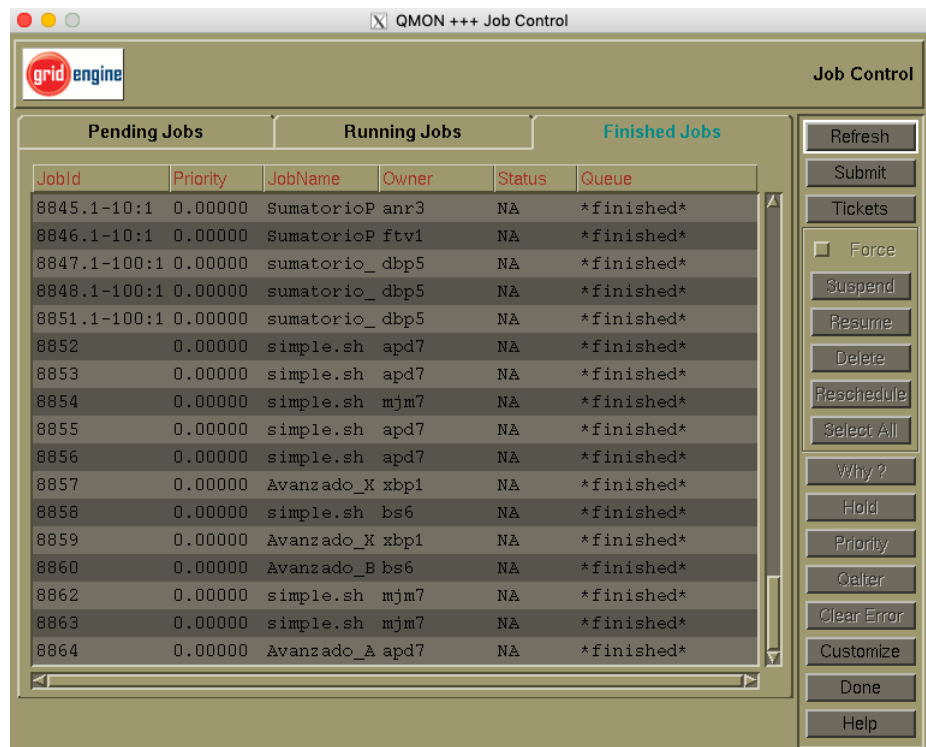
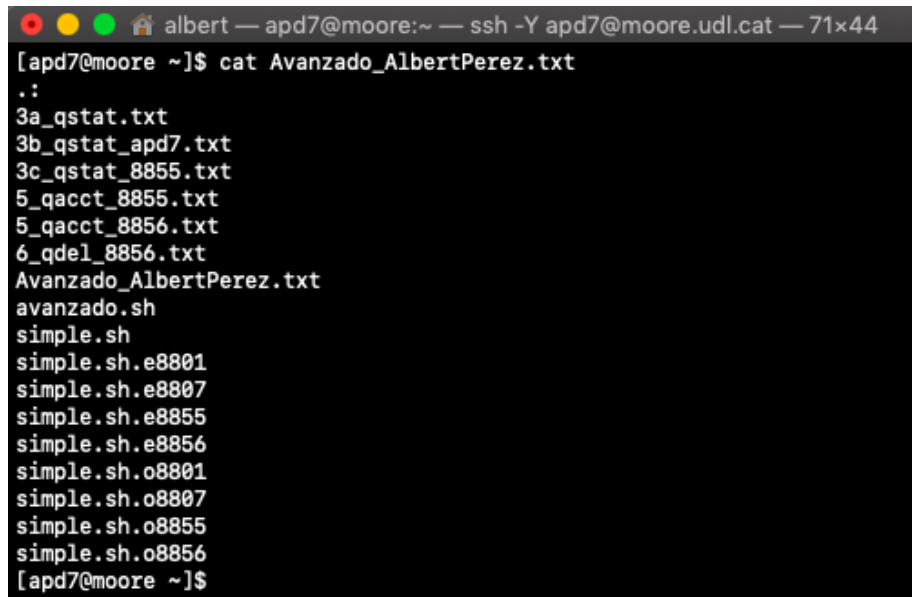
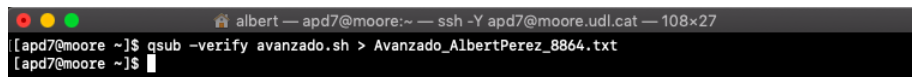


Figure 7: Trabajo 8864 finalizado



```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 71x44
[apd7@moore ~]$ cat Avanzado_AlbertPerez.txt
.:
3a_qstat.txt
3b_qstat_apd7.txt
3c_qstat_8855.txt
5_qacct_8855.txt
5_qacct_8856.txt
6_qdel_8856.txt
Avanzado_AlbertPerez.txt
avanzado.sh
simple.sh
simple.sh.e8801
simple.sh.e8807
simple.sh.e8855
simple.sh.e8856
simple.sh.o8801
simple.sh.o8807
simple.sh.o8855
simple.sh.o8856
[apd7@moore ~]$
```

Figure 8: Visualización fichero de salida trabajo 8864 (se pueden apreciar ficheros respectivos al ejercicio 3 ya que repetí el ejercicio 2 porque la primera vez las opciones del trabajo las especifiqué en la línea de comandos junto al comando qsub y no en el script cómo se pedía)



```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 108x27
[apd7@moore ~]$ qsub -verify avanzado.sh > Avanzado_AlbertPerez_8864.txt
[apd7@moore ~]$
```

Figure 9: Generación del fichero de verificación del script avanzado.sh

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 108x44
[apd7@moore ~]$ cat Avanzado_AlbertPerez_8864.txt
job_number:      unassigned
owner:
uid:             0
group:
gid:             0
sge_o_home:      /home/apd7
sge_o_log_name:  apd7
sge_o_path:      /opt/mpich3/gnu/bin:/usr/lib64/qt-3.3/bin:/usr/local/bin:/usr/bin:/usr/local/sbi
n:/usr/sbin:/opt/ganglia/bin:/opt/ganglia/sbin:/opt/pdsh/bin:/opt/rocks/bin:/opt/rocks/sbin:/opt/gridengine/
bin/lx-amd64:/home/apd7/.local/bin:/home/apd7/bin
sge_o_shell:     /bin/bash
sge_o_workdir:   /home/apd7
sge_o_host:      moore
cwd:             /home/apd7
path_aliases:    /tmp_mnt/ * * /,/private/var/automount/ * * /
merge:          y
hard_resource_list: h_vmem=256M
mail_options:    abe
mail_list:      apd7@alumnes.udl.cat
notify:         FALSE
job_name:        Avanzado_AlbertPerez
stdout_path_list: NONE:NONE:Avanzado_AlbertPerez.txt
jobshare:        0
hard_queue_list: all.q
verify:         -verify
env_list:        TERM
script_size:     159
script_file:     avanzado.sh
script_ptr:
#!/bin/sh
#
##$ -N Avanzado_AlbertPerez
##$ -cwd
##$ -j y
##$ -o Avanzado_AlbertPerez.txt
##$ -q all.q
##$ -l h_vmem=256M
##$ -M apd7@alumnes.udl.cat
##$ -m bea

ls -R
job_type:        NONE
[apd7@moore ~]$
```

Figure 10: Visualización del fichero de verificación del script avanzado.sh

2.2 Ejercicio 2b

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 106x20
[apd7@moore ~]$ qsub avanzado2.sh
Your job 8866 ("Avanzado2_AlbertPerez") has been submitted
[apd7@moore ~]$
```

Figure 11: Lanzamos el trabajo avanzado2.sh con id 8866

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 114x17
[apd7@moore ~]$ qstat -u apd7
job-ID prior  name      user      state submit/start at    queue          slots ja-task-ID
-----
 8866  0.50500 Avanzado2_ apd7      qw    04/19/2020 20:50:51          1              1
[apd7@moore ~]$
```

Figure 12: Observamos que el trabajo se añadió correctamente a la cola y que su estado es "qw" === "esperando"

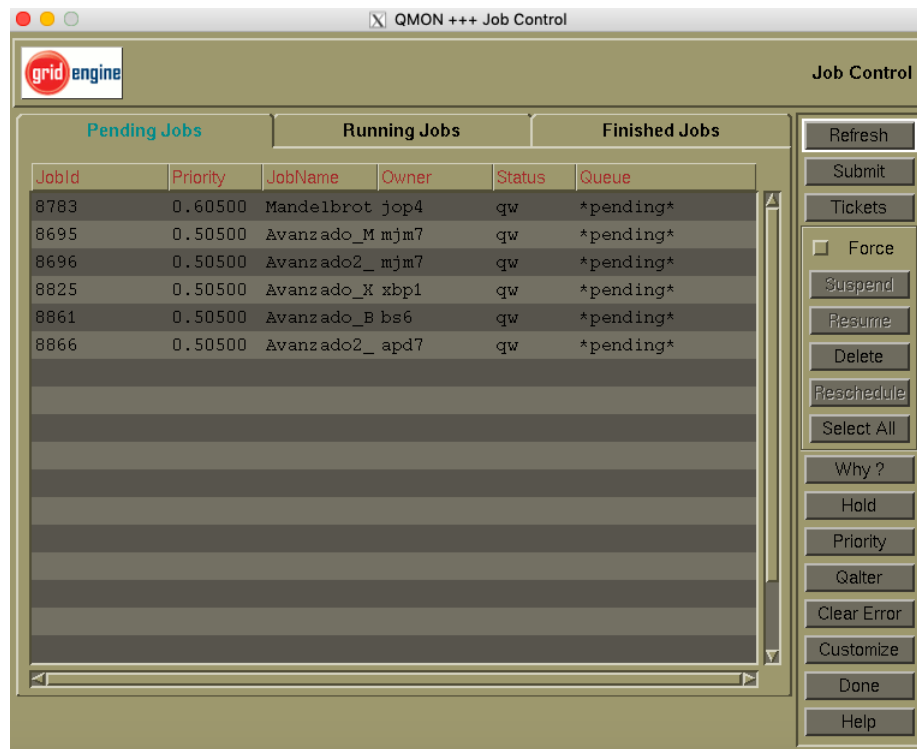


Figure 13: Obervamos mediante la interfaz "qmon" cómo el trabajo está en "Trabajos Pendientes"


```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 80x55
[apd7@moore ~]$ qconf -sq all.q
qname                all.q
hostlist              @allhosts
seq_no                0
load_thresholds       np_load_avg=1.75
suspend_thresholds    NONE
nsuspend              1
suspend_interval      00:05:00
priority              0
min_cpu_interval      00:05:00
processors            UNDEFINED
qtype                 BATCH INTERACTIVE
ckpt_list             NONE
pe_list               make mpi mpich mpich-smp orte smp
rerun                 FALSE
slots                 1,[compute-0-0.local=4],[compute-0-1.local=4], \
                     [compute-0-3.local=4],[compute-0-4.local=4], \
                     [compute-0-5.local=4],[compute-0-6.local=4], \
                     [compute-0-7.local=4],[compute-0-2.local=4]
tmpdir                /tmp
shell                 /bin/sh
prolog                NONE
epilog                NONE
shell_start_mode      posix_compliant
starter_method         NONE
suspend_method         NONE
resume_method         NONE
terminate_method       NONE
notify                00:00:60
owner_list             NONE
user_lists            NONE
xuser_lists           NONE
subordinate_list       NONE
complex_values         NONE
projects              NONE
xprojects             NONE
calendar              NONE
initial_state          default
s_rt                  04:00:00
h_rt                  04:00:00
s_cpu                 INFINITY
h_cpu                 INFINITY
s_fsize               INFINITY
h_fsize               INFINITY
s_data                1g
h_data                1G
s_stack               INFINITY
h_stack               INFINITY
s_core                INFINITY
h_core                INFINITY
s_rss                 INFINITY
h_rss                 INFINITY
s_vmem                1g
h_vmem                1G
[apd7@moore ~]$
```

Figure 14: Visualización de la configuración de la cola "all.q" (Observamos que nuestro trabajo nunca se ejecutará porque supera el límite "h_rt=04:00:00" establecido)

3 Ejercicio 3: Monitorización Trabajos

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 108x24
[apd7@moore ~]$ qsub simple.sh
Your job 8855 ("simple.sh") has been submitted
[apd7@moore ~]$ qstat > 3a_qstat.txt
[apd7@moore ~]$ ls
3a_qstat.txt  outAvanzado_AlbertPerez.txt  simple.sh.e8807  simple.sh.o8807
avanzado.sh  simple.sh                  simple.sh.e8855  simple.sh.o8855
avanzado2.sh simple.sh.e8801            simple.sh.o8801
[apd7@moore ~]$ qstat -u apd7 > 3b_qstat_apd7.txt
[apd7@moore ~]$ qstat -j 8855 > 3c_qstat_8855.txt
[apd7@moore ~]$
```

Figure 15: Lanzamos el trabajo simple.sh con id 8855, y generamos ficheros de volcado a partir del comando de monitorización "qstat" y sus diversas opciones para saber el estado de nuestro trabajo.

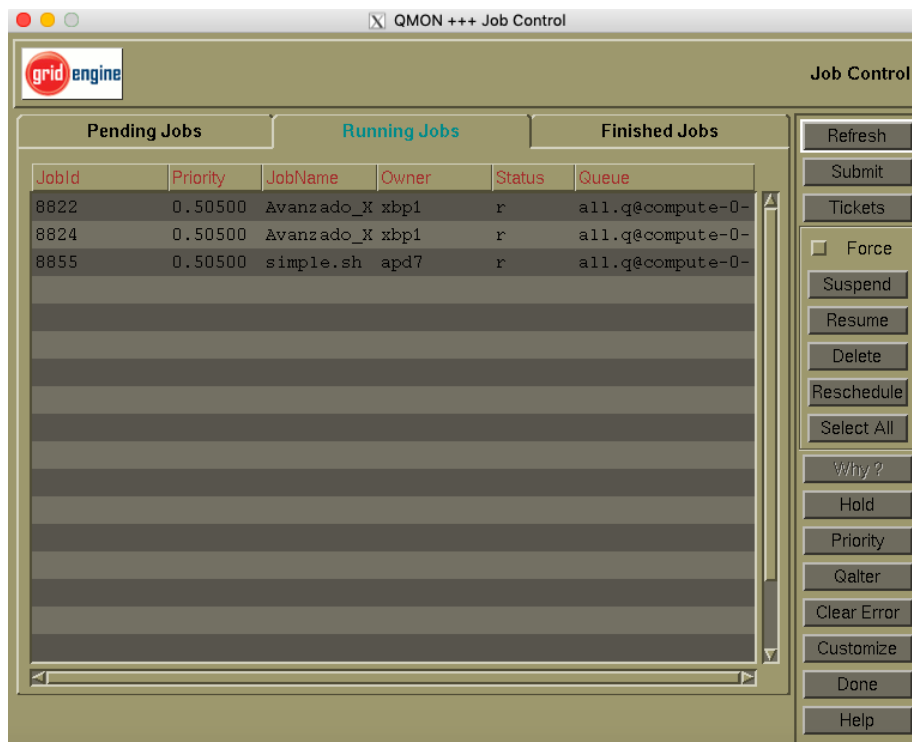


Figure 16: Mediante "qmon" monitorizamos el trabajo 8855, y observamos que se está ejecutando

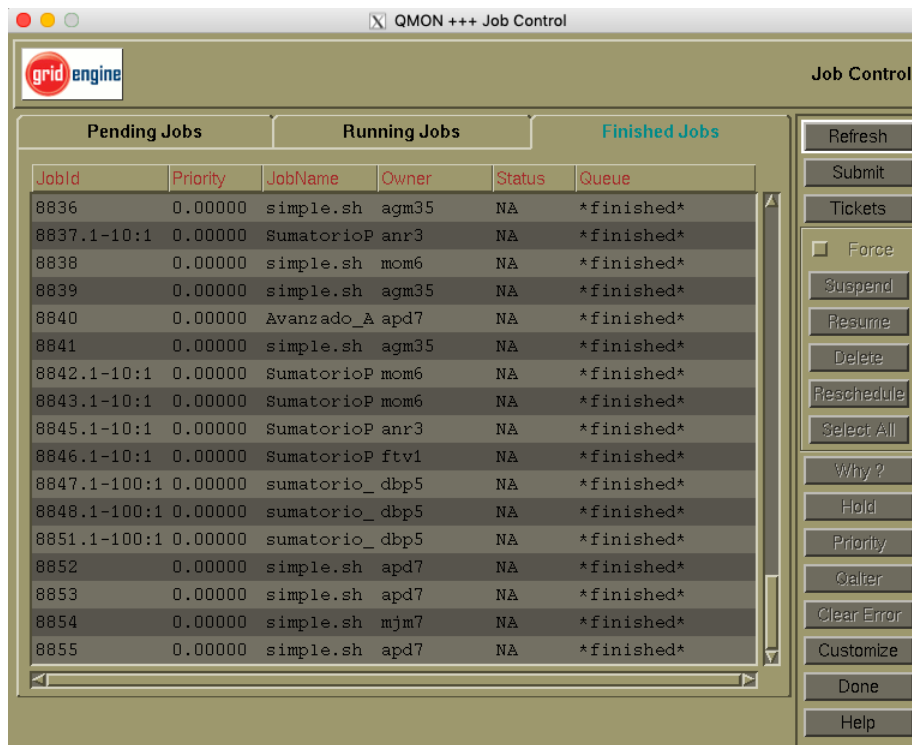


Figure 17: Mediante "qmon" observamos que el trabajo 8855 ha finalizado

```

[apd7@moore ~]$ qacct -j 8855 > 5_qacct_8855.txt
[apd7@moore ~]$ ls
3a_qstat.txt      5_qacct_8855.txt  outAvanzado_AlbertPerez.txt  simple.sh.e8807  simple.sh.e8807
3b_qstat_apd7.txt avanzado.sh        simple.sh                    simple.sh.e8855  simple.sh.e8855
3c_qstat_8855.txt avanzado2.sh       simple.sh.e8801              simple.sh.e8801
[apd7@moore ~]$ cat 5_qacct_8855.txt
=====
qname      all.q
hostname   compute-0-7.local
group      cda
owner      apd7
project    NONE
department defaultdepartment
jobname    simple.sh
jobnumber  8855
taskid     undefined
account    sge
priority   0
qsub_time  Sun Apr 19 19:37:27 2020
start_time Sun Apr 19 19:37:33 2020
end_time   Sun Apr 19 19:39:33 2020
granted_pe NONE

```

Generamos fichero de volcado a partir del comando "qacct" especificando con la opción "-j" el trabajo 8855 para saber su utilización de recursos. Después visualizamos el fichero generado

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 108x24
[apd7@moore ~]$ qsub simple.sh
Your job 8856 ("simple.sh") has been submitted
[apd7@moore ~]$ qdel 8856 > 6_qdel_8856.txt
[apd7@moore ~]$ qacct -j 8856 > 5_qacct_8856.txt
[apd7@moore ~]$
```

Figure 18: Lanzamos el trabajo simple.sh con id 8856. Lo eliminamos cuando se está ejecutando (lo sabemos gracias la siguiente captura). Y generamos fichero de utilización de recursos.

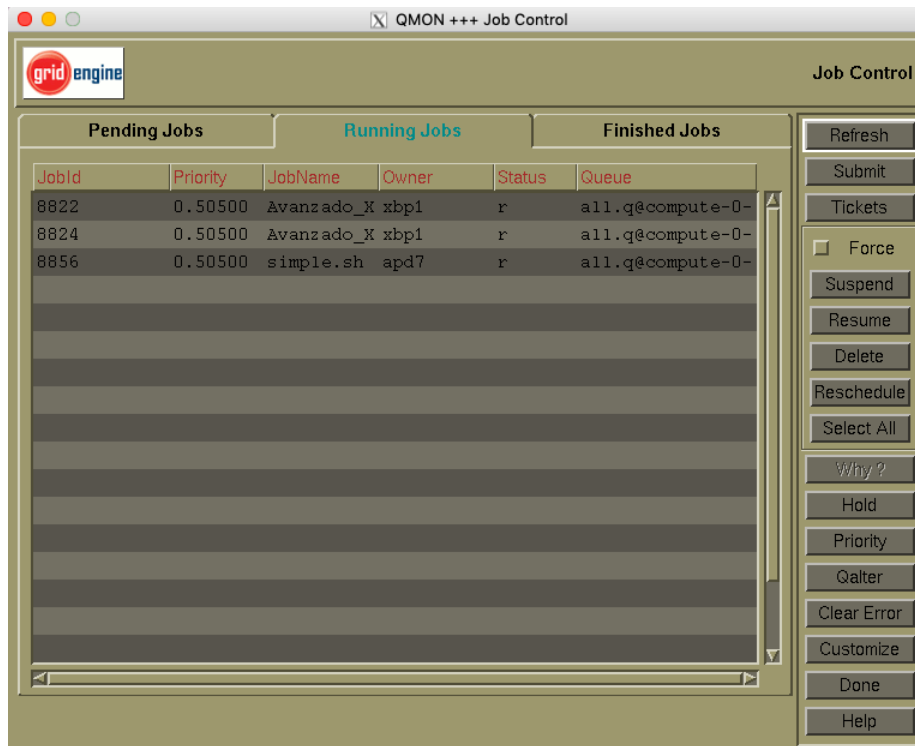


Figure 19: Observamos mediante "qmon" que nuestro trabajo ya se está ejecutando. Entonces ya podemos eliminarlo usando el comando "qdel" y especificando el id de dicho trabajo

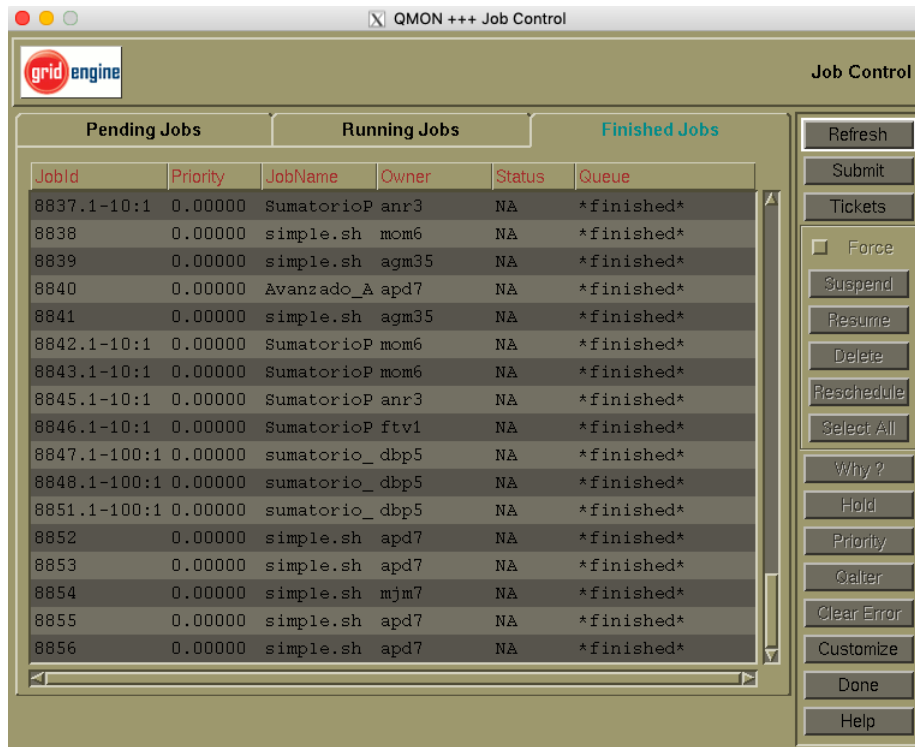


Figure 20: Trabajo 8856 finalizado

4 Ejercicio 4: Matriz de Tareas

```
#!/bin/bash
#$ -S /bin/bash
#$ -N SumatorioPthreads_apd7
#$ -cwd
#$ -t 1-100 #generación 100 tareas
#$ -j y
#$ -o outputTareas.txt

M=$((($SGE_TASK_ID+9)/10))
N=$((($SGE_TASK_ID%10))

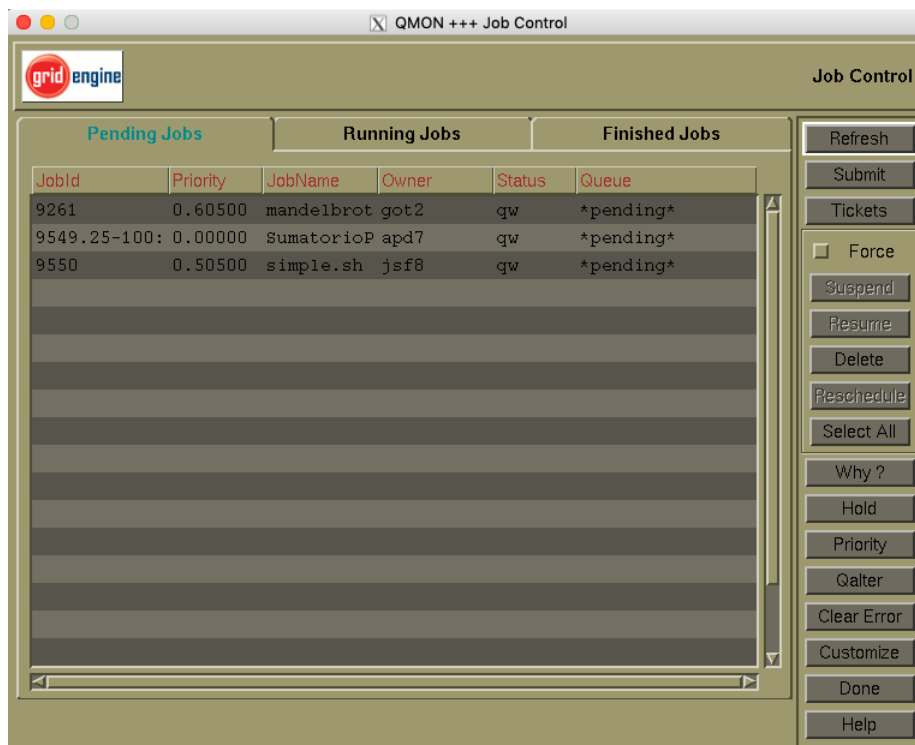
if [ "$N" -eq "0" ]; then
    N=10
fi;

declare -a array=({ time ./SumatorioPthreads $((M*10+8)) $N; } 2>&1 >/dev/null )
echo "Tarea $SGE_TASK_ID: Sumatorio $M*10*8 primeros numeros con $N threads: ${array[*]}" >> outMatrizTareas.txt
```

Figure 21: Script matrizTareas.sh

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 102x23
[apd7@moore ~]$ qsub matrizTareas.sh
Your job-array 9549.1-100:1 ("SumatorioPthreads_apd7") has been submitted
[apd7@moore ~]$
```

Figure 22: Lanzamos el trabajo matrizTareas.sh con id 9549



Captura monitorización. Tanto en esta como en las siguiente vemos como se nos indica la siguiente tarea a entrar en ejecución, en este caso seria la 25

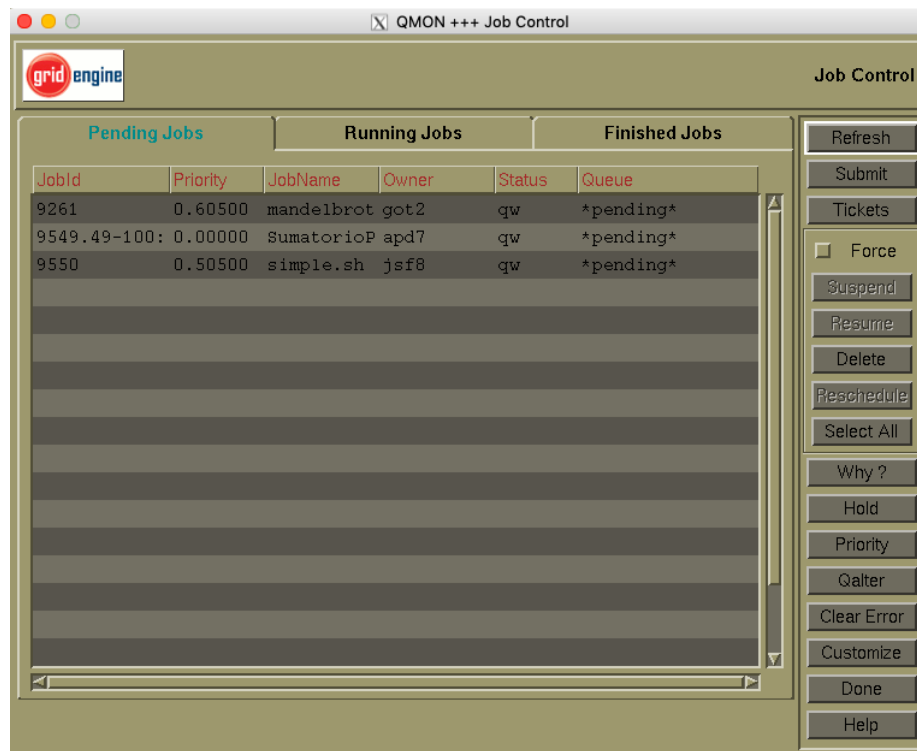


Figure 23: Captura monitorización

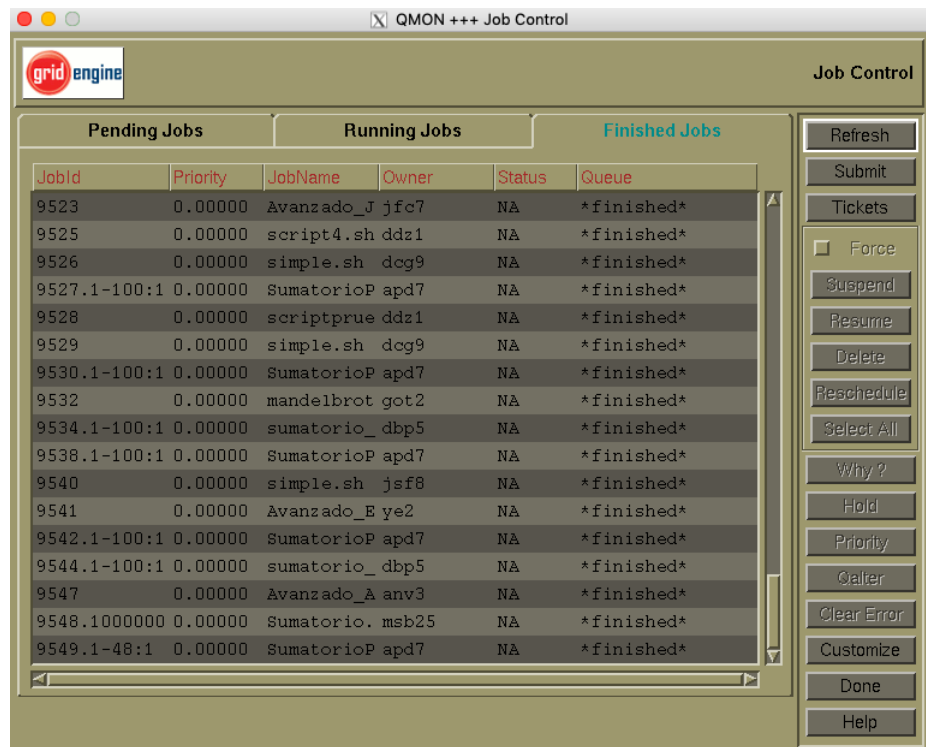


Figure 24: Captura monitorización

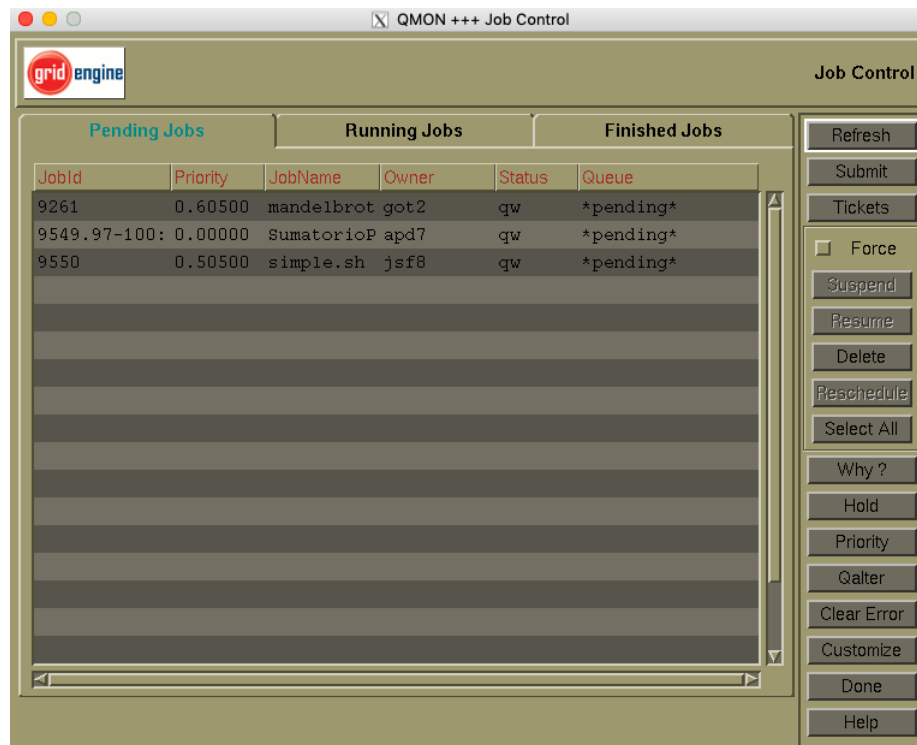


Figure 25: Captura monitorización

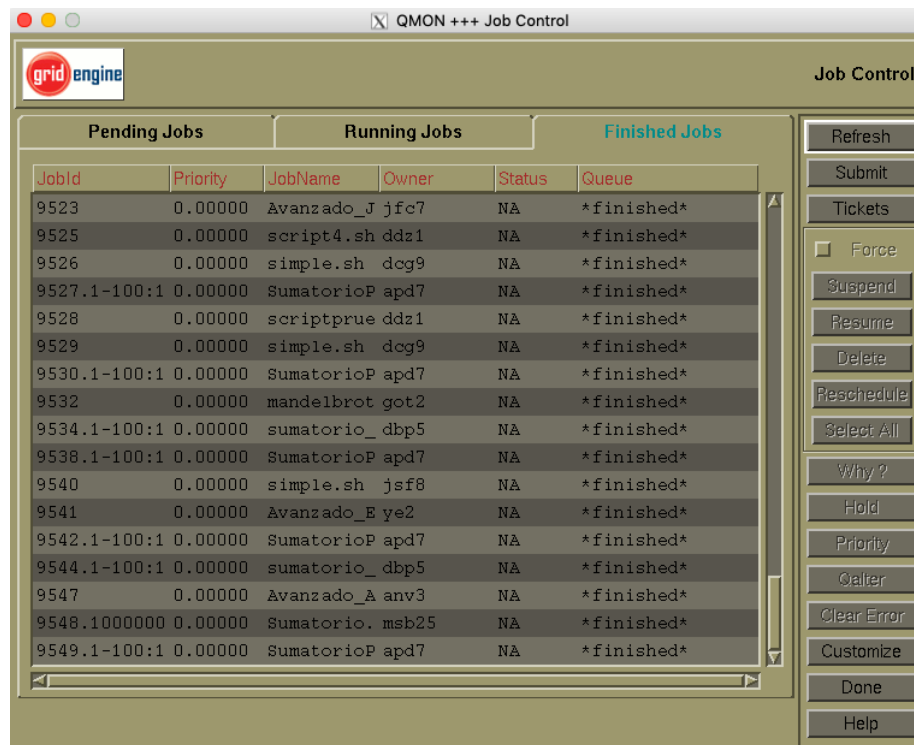


Figure 26: Captura monitorización (trabajo 9549 finalizado al completo)

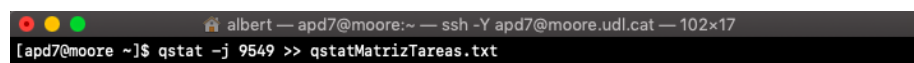


Figure 27: Generación fichero de monitorización de estado del trabajo. Durante la ejecución del trabajo, lanzamos este comando para capturar su estado y conseguir tener más datos de monitorización

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 103x17
[apd7@moore ~]$ cat qstatMatrizTareas.txt
=====
job_number:          9549
exec_file:           job_scripts/9549
submission_time:     Wed Apr 22 17:11:15 2020
owner:               apd7
uid:                 1068
group:               cda
gid:                 1028
sge_o_home:          /home/apd7
sge_o_log_name:      apd7
sge_o_path:           /opt/mpich3/gnu/bin:/usr/lib64/qt-3.3/bin:/usr/local/bin:/usr/bin:/usr/loca
l/sbin:/usr/sbin:/opt/ganglia/bin:/opt/ganglia/sbin:/opt/pdsh/bin:/opt/rocks/bin:/opt/rocks/sbin:/opt/g
ridengine/bin/lx-amd64:/home/apd7/.local/bin:/home/apd7/bin
sge_o_shell:         /bin/bash
sge_o_workdir:        /home/apd7
sge_o_host:          moore
```

Figure 28: Visualización fichero monitorización del estado del trabajo 9549

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 102x17
[apd7@moore ~]$ qacct -j 9549 >> qacctMatrizTareas.txt
[apd7@moore ~]$
```

Figure 29: Generación fichero de información adicional sobre el trabajo 9549 ya finalizado. En este caso veremos información acerca de cada una de las tareas ejecutadas

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 103x17
[apd7@moore ~]$ cat qacctMatrizTareas.txt
=====
qname      all.q
hostname   compute-0-2.local
group      cda
owner      apd7
project    NONE
department defaultdepartment
jobname     SumatorioPthreads_apd7
jobnumber  9549
taskid     1
account    sge
priority   0
qsub_time  Wed Apr 22 17:11:15 2020
start_time Wed Apr 22 17:11:25 2020
end_time   Wed Apr 22 17:11:25 2020
granted_pe NONE
```

Figure 30: Visualización fichero de información adicional

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 103x17
[apd7@moore ~]$ cat outputTareas.txt
[apd7@moore ~]$
```

Figure 31: Visualización Fichero de salida del trabajo

```

albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 103x17
[apd7@moore ~]$ cat outMatrizTareas.txt
Tarea 4: Sumatorio 1*10^8 primeros numeros con 4 threads: real 0m0.078s user 0m0.264s sys 0m0.001s
Tarea 5: Sumatorio 1*10^8 primeros numeros con 5 threads: real 0m0.104s user 0m0.254s sys 0m0.000s
Tarea 6: Sumatorio 1*10^8 primeros numeros con 6 threads: real 0m0.093s user 0m0.262s sys 0m0.000s
Tarea 7: Sumatorio 1*10^8 primeros numeros con 7 threads: real 0m0.108s user 0m0.253s sys 0m0.001s
Tarea 3: Sumatorio 1*10^8 primeros numeros con 3 threads: real 0m0.147s user 0m0.260s sys 0m0.001s
Tarea 8: Sumatorio 1*10^8 primeros numeros con 8 threads: real 0m0.095s user 0m0.254s sys 0m0.000s
Tarea 10: Sumatorio 1*10^8 primeros numeros con 10 threads: real 0m0.083s user 0m0.254s sys 0m0.000s
Tarea 2: Sumatorio 1*10^8 primeros numeros con 2 threads: real 0m0.206s user 0m0.260s sys 0m0.000s
Tarea 9: Sumatorio 1*10^8 primeros numeros con 9 threads: real 0m0.076s user 0m0.253s sys 0m0.001s
Tarea 17: Sumatorio 2*10^8 primeros numeros con 7 threads: real 0m0.159s user 0m0.503s sys 0m0.003s
Tarea 1: Sumatorio 1*10^8 primeros numeros con 1 threads: real 0m0.390s user 0m0.259s sys 0m0.000s
Tarea 15: Sumatorio 2*10^8 primeros numeros con 5 threads: real 0m0.156s user 0m0.502s sys 0m0.002s
Tarea 16: Sumatorio 2*10^8 primeros numeros con 6 threads: real 0m0.149s user 0m0.505s sys 0m0.001s
Tarea 18: Sumatorio 2*10^8 primeros numeros con 8 threads: real 0m0.170s user 0m0.506s sys 0m0.000s
Tarea 19: Sumatorio 2*10^8 primeros numeros con 9 threads: real 0m0.174s user 0m0.505s sys 0m0.001s
Tarea 13: Sumatorio 2*10^8 primeros numeros con 3 threads: real 0m0.280s user 0m0.500s sys 0m0.001s

```

Figure 32: Visualización fichero de salida de las tareas. Aquí podemos ver el resultado de los tiempos

5 Ejercicio 5: Trabajos Paralelos

5.1 Ejercicio 5a

```

albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 99x25
[apd7@moore ~]$ time ./SumatorioPthreads 100000000 1
Sumatorio 100000000 primeros numeros: 5000000050000000 (resultado esperado 5000000050000000).

real    0m4.787s
user    0m4.785s
sys      0m0.002s
[apd7@moore ~]$ time ./SumatorioPthreads 100000000 2
Sumatorio 100000000 primeros numeros: 5000000050000000 (resultado esperado 5000000050000000).

real    0m2.402s
user    0m4.796s
sys      0m0.002s
[apd7@moore ~]$ time ./SumatorioPthreads 100000000 3
Sumatorio 100000000 primeros numeros: 5000000050000000 (resultado esperado 5000000050000000).

real    0m1.604s
user    0m4.802s
sys      0m0.002s
[apd7@moore ~]$ time ./SumatorioPthreads 100000000 4
Sumatorio 100000000 primeros numeros: 5000000050000000 (resultado esperado 5000000050000000).

real    0m1.206s
user    0m4.806s
sys      0m0.003s
[apd7@moore ~]$

```

Figure 33: Tiempos del SumatorioPthreads en el front-end, variando los threads de 1 a 4

```
Tarea 91: Sumatorio 10*10^8 primeros numeros con 1 threads: real 0m2.624s user 0m2.431s sys 0m0.000s
Tarea 92: Sumatorio 10*10^8 primeros numeros con 2 threads: real 0m2.568s user 0m2.492s sys 0m0.000s
Tarea 93: Sumatorio 10*10^8 primeros numeros con 3 threads: real 0m1.249s user 0m2.486s sys 0m0.001s
Tarea 94: Sumatorio 10*10^8 primeros numeros con 4 threads: real 0m0.967s user 0m2.520s sys 0m0.002s
```

Figure 34: Tiempos de las tareas equivalentes (datos obtenidos del fichero out-MatrizTareas.txt)

Vemos que el speedup, la mejora del rendimiento (en este caso lo medimos con tiempos) desde usar 1 thread a usar 4 threads, es bastante más grande en el front-end que en el cluster.

Es decir, en el front-end pasamos de 4 segundos aprox. a 1 segundo. I en el cluster de 2 segundos y algo a casi 1 segundo. Hay bastante diferencia de mejora.

Esto es debido a que ejecutamos el trabajo sin usar la potencia de cómputo que nos proporciona el cluster. Ya que solamente estamos usando 1 slot. (Se puede ver mediante el comando qstat o a la configuración de la cola all.q usando qconf -sq all.q veremos que hay 1 slot de forma predeterminada)

Si deseamos mejorar el rendimiento entonces, deberíamos optar por usar un entorno paralelo y aumentar el número de slots.

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 104x44
[apd7@moore ~]$ qsub matrizTareasParallel.sh
Your job-array 9685.1-100:1 ("ParallelSumatorio_apd7") has been submitted
[apd7@moore ~]$
```

Figure 35: Lanzamos trabajo matrizTareasParallel.sh con id 9685

```

[apd7@moore ~]$ qstat
job-ID prior      name          user      state submit/start at   queue                          slots ja-task-ID
-----
 9685 0.52500 ParallelSu apd7      qw    04/22/2020 20:38:19      4 69-100:1
[apd7@moore ~]$ qstat
job-ID prior      name          user      state submit/start at   queue                          slots ja-task-ID
-----
 9685 0.52500 ParallelSu apd7      qw    04/22/2020 20:38:19      4 69-100:1
[apd7@moore ~]$ qstat
job-ID prior      name          user      state submit/start at   queue                          slots ja-task-ID
-----
 9685 0.52500 ParallelSu apd7      qw    04/22/2020 20:38:19      4 75-100:1
[apd7@moore ~]$ qstat
job-ID prior      name          user      state submit/start at   queue                          slots ja-task-ID
-----
 9685 0.52500 ParallelSu apd7      qw    04/22/2020 20:38:19      4 79-100:1
[apd7@moore ~]$ qstat
job-ID prior      name          user      state submit/start at   queue                          slots ja-task-ID
-----
 9685 0.52500 ParallelSu apd7      qw    04/22/2020 20:38:19      4 85-100:1
[apd7@moore ~]$ qstat
job-ID prior      name          user      state submit/start at   queue                          slots ja-task-ID
-----
 9685 0.52500 ParallelSu apd7      qw    04/22/2020 20:38:19      4 88-100:1
[apd7@moore ~]$
[apd7@moore ~]$

```

Figure 36: Captura monitorización usando "qstat"

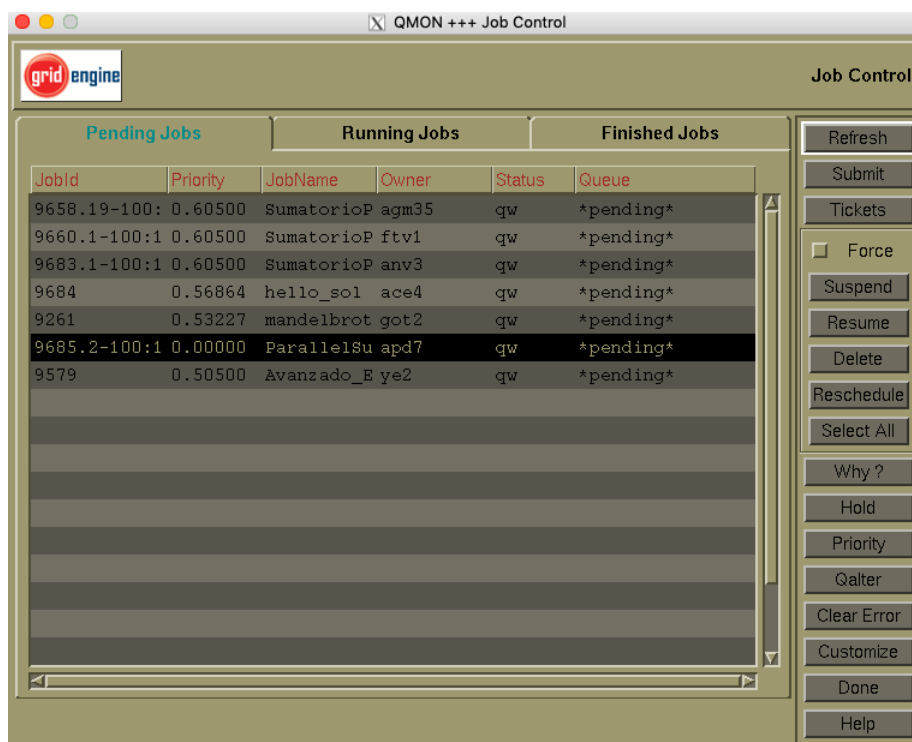


Figure 37: Captura monitorización

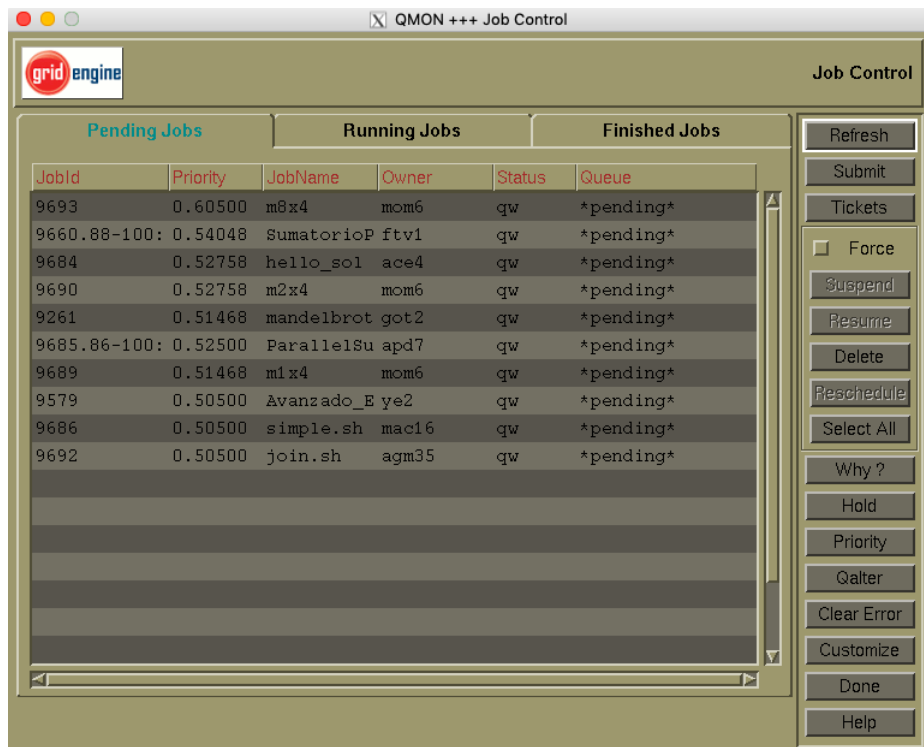


Figure 38: Captura monitorización

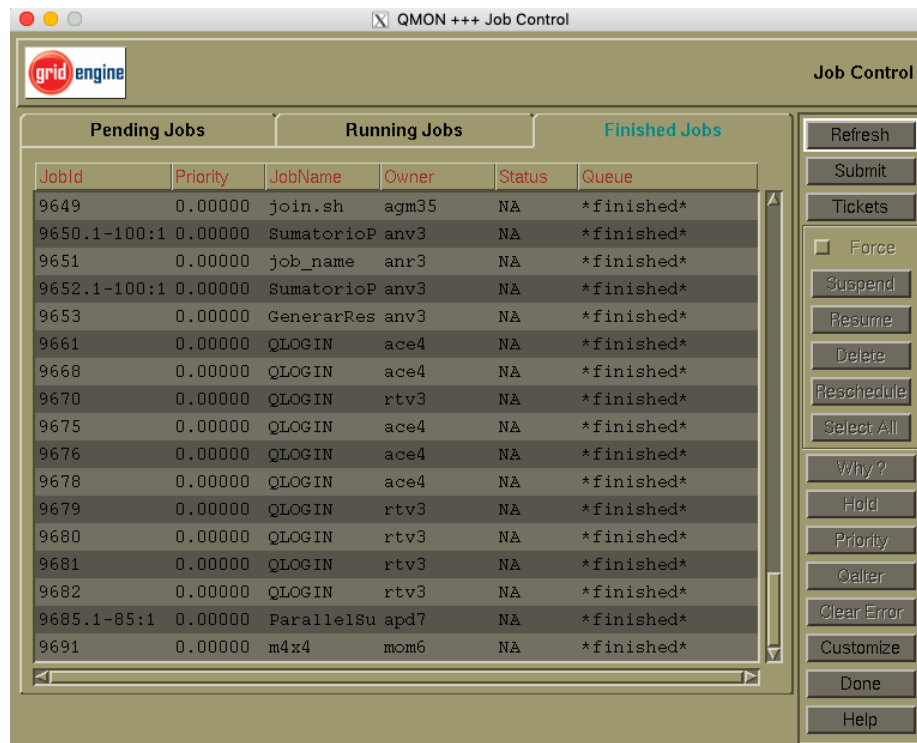


Figure 39: Captura monitorización

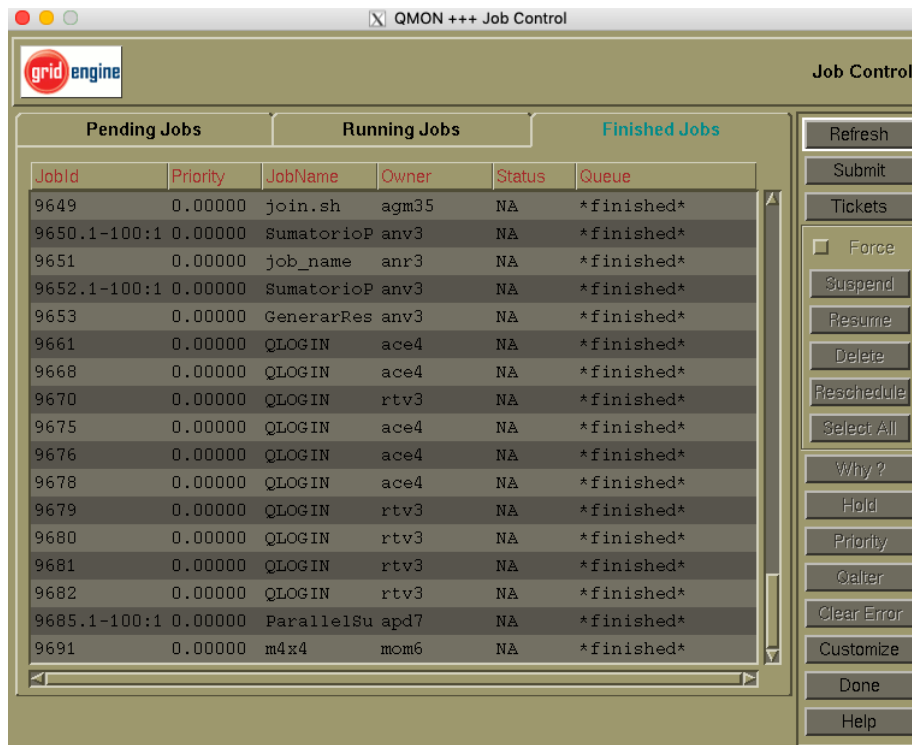


Figure 40: Captura monitorización. Trabajo 9685 finalizado

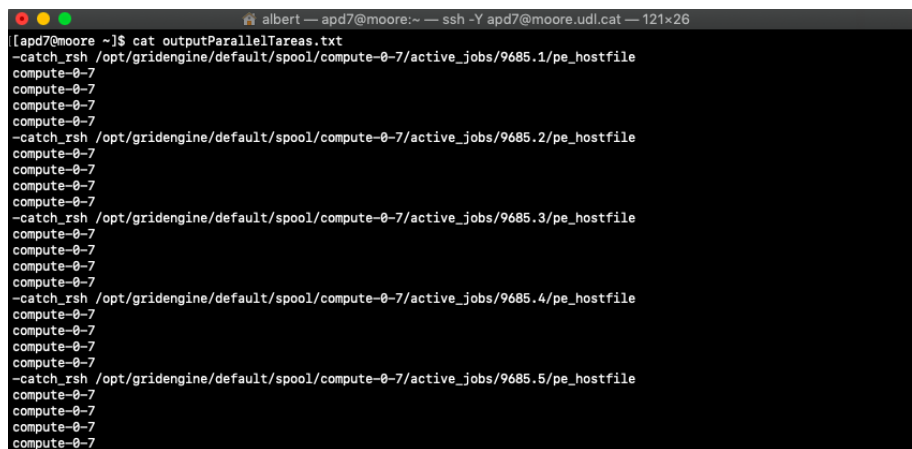


Figure 41: Visualización fichero salida outputParallelTareas.txt

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 121x26
[apd7@moore ~]$ cat outParallelMatrizTareas.txt
Tarea 1: Sumatorio 1*10^8 primeros numeros con 1 threads: real 0m0.246s user 0m0.242s sys 0m0.002s
Tarea 2: Sumatorio 1*10^8 primeros numeros con 2 threads: real 0m0.128s user 0m0.254s sys 0m0.000s
Tarea 3: Sumatorio 1*10^8 primeros numeros con 3 threads: real 0m0.087s user 0m0.258s sys 0m0.000s
Tarea 4: Sumatorio 1*10^8 primeros numeros con 4 threads: real 0m0.070s user 0m0.262s sys 0m0.001s
Tarea 5: Sumatorio 1*10^8 primeros numeros con 5 threads: real 0m0.081s user 0m0.259s sys 0m0.000s
Tarea 6: Sumatorio 1*10^8 primeros numeros con 6 threads: real 0m0.072s user 0m0.259s sys 0m0.000s
Tarea 7: Sumatorio 1*10^8 primeros numeros con 7 threads: real 0m0.075s user 0m0.259s sys 0m0.002s
Tarea 8: Sumatorio 1*10^8 primeros numeros con 8 threads: real 0m0.067s user 0m0.264s sys 0m0.000s
Tarea 9: Sumatorio 1*10^8 primeros numeros con 9 threads: real 0m0.071s user 0m0.260s sys 0m0.000s
Tarea 10: Sumatorio 1*10^8 primeros numeros con 10 threads: real 0m0.068s user 0m0.263s sys 0m0.001s
Tarea 11: Sumatorio 2*10^8 primeros numeros con 1 threads: real 0m0.483s user 0m0.482s sys 0m0.001s
Tarea 12: Sumatorio 2*10^8 primeros numeros con 2 threads: real 0m0.250s user 0m0.498s sys 0m0.001s
Tarea 13: Sumatorio 2*10^8 primeros numeros con 3 threads: real 0m0.168s user 0m0.502s sys 0m0.000s
Tarea 14: Sumatorio 2*10^8 primeros numeros con 4 threads: real 0m0.133s user 0m0.512s sys 0m0.000s
Tarea 15: Sumatorio 2*10^8 primeros numeros con 5 threads: real 0m0.155s user 0m0.512s sys 0m0.001s
Tarea 16: Sumatorio 2*10^8 primeros numeros con 6 threads: real 0m0.142s user 0m0.523s sys 0m0.000s
Tarea 17: Sumatorio 2*10^8 primeros numeros con 7 threads: real 0m0.150s user 0m0.520s sys 0m0.001s
Tarea 18: Sumatorio 2*10^8 primeros numeros con 8 threads: real 0m0.134s user 0m0.524s sys 0m0.000s
Tarea 19: Sumatorio 2*10^8 primeros numeros con 9 threads: real 0m0.139s user 0m0.516s sys 0m0.000s
Tarea 20: Sumatorio 2*10^8 primeros numeros con 10 threads: real 0m0.138s user 0m0.515s sys 0m0.001s
Tarea 21: Sumatorio 3*10^8 primeros numeros con 1 threads: real 0m0.722s user 0m0.721s sys 0m0.000s
Tarea 22: Sumatorio 3*10^8 primeros numeros con 2 threads: real 0m0.372s user 0m0.741s sys 0m0.000s
Tarea 23: Sumatorio 3*10^8 primeros numeros con 3 threads: real 0m0.251s user 0m0.740s sys 0m0.000s
Tarea 24: Sumatorio 3*10^8 primeros numeros con 4 threads: real 0m0.192s user 0m0.765s sys 0m0.000s
Tarea 25: Sumatorio 3*10^8 primeros numeros con 5 threads: real 0m0.233s user 0m0.765s sys 0m0.000s
```

Figure 42: Visualización fichero salida outParallelMatrizTareas.txt

```
Tarea 91: Sumatorio 10*10^8 primeros numeros con 1 threads: real 0m2.393s user 0m2.393s sys 0m0.000s
Tarea 92: Sumatorio 10*10^8 primeros numeros con 2 threads: real 0m1.236s user 0m2.467s sys 0m0.001s
Tarea 93: Sumatorio 10*10^8 primeros numeros con 3 threads: real 0m0.821s user 0m2.457s sys 0m0.001s
Tarea 94: Sumatorio 10*10^8 primeros numeros con 4 threads: real 0m0.635s user 0m2.535s sys 0m0.000s
```

Figure 43: Usando el entorno paralelo, en este caso con 4 slots, observamos en el fichero outParallelMatrizTareas.txt cómo el speedup es mejor

5.2 Ejercicio 5b

En este ejercicio mhe ha resultado complicado ejectuar los trabajos debido a la cantidad de slots requeridos. Esto me ha surgido ya que el cluster está un poco saturado.

He intentado ejectuar el script con los diferentes slots (de 4 a 32). En las pruebas que hice al principio, la ejecución con 4, 8 y 16 me funcionaron. Pero la de 32 no. Entonces contacté con el profesor, para explicarle la situación.

Desgraciadamente no hice ni la monitorización ni la generación de los ficheros de ejecución de las primeras ejecuciones. Y por eso me vine obligado a repetirlas.

Llevo ya dos días probando de ejecutar, dichos trabajos en el cluster. Pero éste esta bastante saturado. Ni tan sólo me ejecuta el script pidiendo 4 slots.

En las siguientes capturas se ve el lanzamiento del trabajo de 4 slots y su monitorización en la cola de trabajos pendientes. También el lanzamiento con los otros slots requeridos.

Además he generado los ficheros de monitorización. Aunque en ellos se especifica que no se pudo correr el programa debido a que solo habían 'x' slots disponibles. Donde x es más pequeña que los requeridos.

```
#!/bin/sh
#$ -S /bin/bash
#$ -N cpi_apd7_4
#$ -q all.q
#$ -cwd
#$ -j y
#$ -o $JOB_NAME.o$JOB_ID
#$ -pe mpich 4

MPICH_MACHINES=$TMPDIR/mpich_machines
cat $PE_HOSTFILE | awk '{print $1":"$2}' > $MPICH_MACHINES

echo $MPICH_MACHINES
echo $NSLOTS

## In this line you have to write the command that will execute your application.
time mpiexec -f $MPICH_MACHINES -n $NSLOTS ./cpi

rm -rf $MPICH_MACHINES
```

Figure 44: Script del trabajo. En este específico caso se usan 4 slots.

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 80x24
[apd7@moore ~]$ qsub cpi.sh
Your job 10264 ("cpi_apd7_4") has been submitted
[apd7@moore ~]$ qstat
job-ID prior name user state submit/start at queue
slots ja-task-ID
-----
10264 0.00000 cpi_apd7_4 apd7 qw 04/24/2020 22:16:23
4
[apd7@moore ~]$ qstat -j 10264 >> qstat_cpi_4.txt
[apd7@moore ~]$
```

Figure 45: Trabajo cpi.sh con 4 slots lanzado con id 10264. Visualización estado mediante 'qstat'. Y generación fichero de monitorización

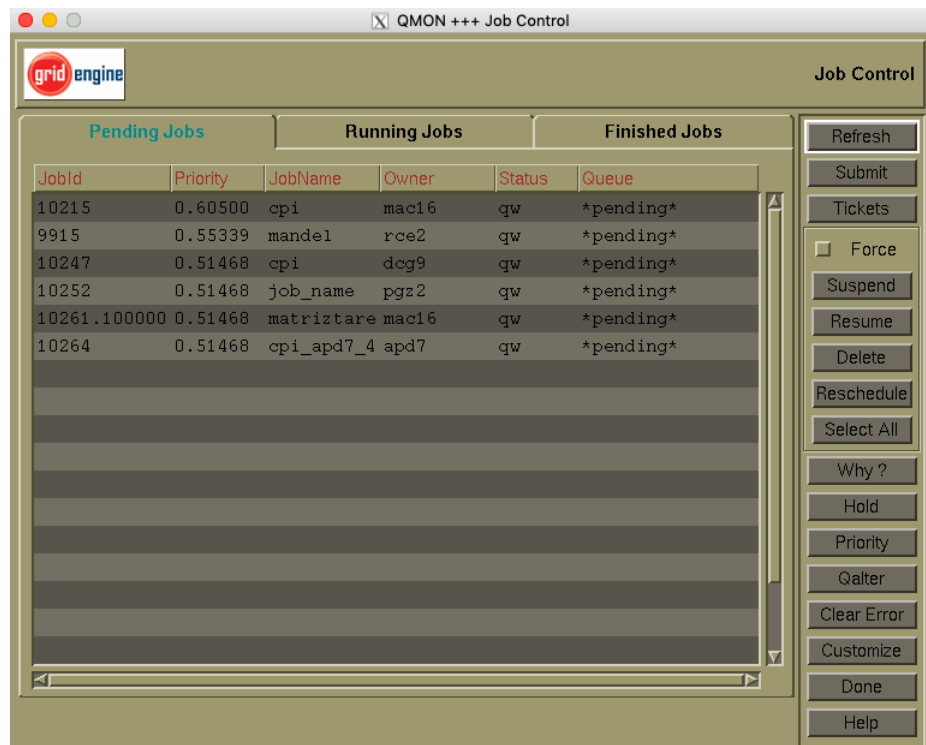


Figure 46: Visualización monitorización mediante 'qmon'. El trabajo ya está en la cola.

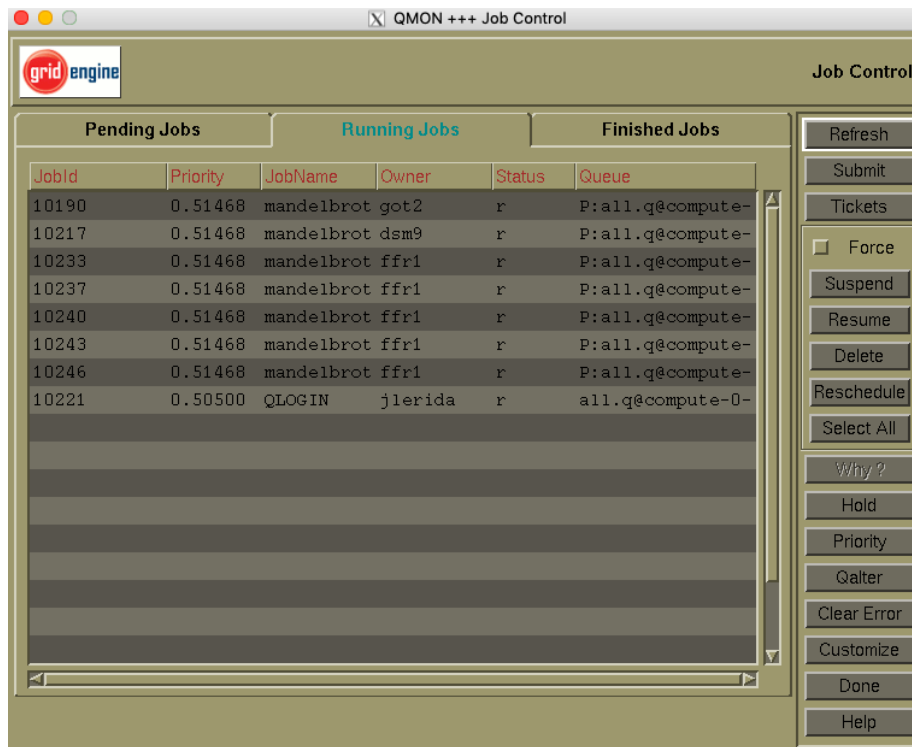


Figure 47: Visualización de los trabajos que se están ejecutando. Cómo podemos ver hay bastante grado de saturación.

```

[apd7@moore ~]$ qsub cpi.sh
Your job 10279 ("cpi_apd7_8") has been submitted
[apd7@moore ~]$ qstat
job-ID prior name user state submit/start at queue
slots ja-task-ID
-----
10279 0.00000 cpi_apd7_8 apd7 qw 04/24/2020 23:21:37
8
[apd7@moore ~]$ qstat -j 10279 >> qstat_cpi_8.txt
[apd7@moore ~]$

```

Figure 48: Trabajo cpi.sh con 8 slots lanzado con id 10279

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 96x24
[apd7@moore ~]$ qsub cpi.sh
Your job 10281 ("cpi_apd7_16") has been submitted
[apd7@moore ~]$ qstat
job-ID prior name user state submit/start at queue
slots ja-task-ID
-----
10281 0.00000 cpi_apd7_1 apd7 qw 04/24/2020 23:24:02
16
[apd7@moore ~]$ qstat -j 10281 >> qstat_cpi_16.txt
[apd7@moore ~]$
```

Figure 49: Trabajo cpi.sh con 16 slots lanzado con id 10281

```
albert — apd7@moore:~ — ssh -Y apd7@moore.udl.cat — 96x24
[apd7@moore ~]$ qsub cpi.sh
Your job 10282 ("cpi_apd7_32") has been submitted
[apd7@moore ~]$ qstat
job-ID prior name user state submit/start at queue
slots ja-task-ID
-----
10282 0.00000 cpi_apd7_3 apd7 qw 04/24/2020 23:26:08
32
[apd7@moore ~]$ qstat -j 10282 >> qstat_cpi_32.txt
[apd7@moore ~]$
```

Figure 50: Trabajo cpi.sh con 32 slots lanzado con id 10282

La idea de éste ejercicio era comparar los rendimientos de cada una de las opciones.

A pesar que a nivel práctico no lo he podido experimentar. Me veo con la obligación de al menos comentar de forma teórica cómo podrían haber sido estos resultados.

En mi opinión, cuántos más slots más potencia de computación ya que estamos usando más procesadores de forma paralela.

Entonces si eleborásemos una gráfica de prestaciones veríamos cómo a medida que aumentasen los slots, el tiempo de ejecución total se reduciría pero sólo hasta un punto.

Este punto depende de las características personales del programa/trabajo. Ya que habría una cantidad de slots a partir de la cuál la mejora del rendimiento se estancaría, ya que un programa se puede paralelizar pero hasta cierto punto. No tiene sentido ejecutar usando 5 slots cuando el programa sólo tiene 4 tareas paralelas para ejecutar.