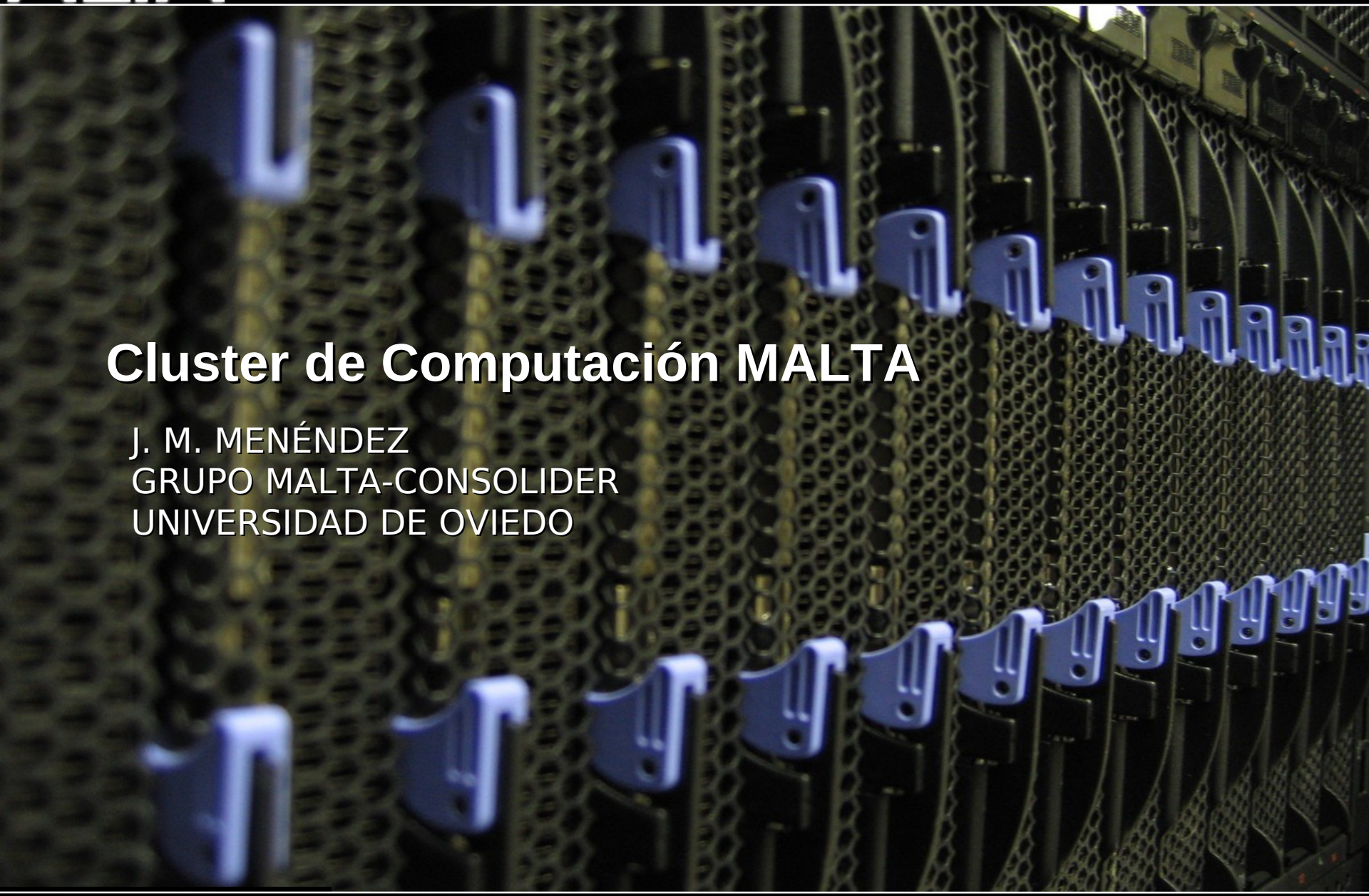


Cluster de Computación MALTA

J. M. MENÉNDEZ
GRUPO MALTA-CONSOLIDER
UNIVERSIDAD DE OVIEDO

Especificaciones Técnicas

(688GB RAM, 7TB)

Chasis 1 – chasis de control

- 1 procesador Intel Quad Core 5450 a 3 Ghz
- 4 GB de RAM
- 2 discos SAS de 146 GB
- 2 interfaces Gigabit

Chasis 2 – I/O Bounded (14 nodos)

- 2 procesador Intel Quad Core 5450 a 3 Ghz
- 16 (4x4) GB de RAM
- 2 discos SAS de 146 GB
- 2 interfaces Gigabit

Chasis 3 – CPU Bounded (14 nodos)

- 2 procesador Intel Quad Core 5450 a 3 Ghz
- 32 (8x4) GB de RAM
- 1 discos SAS de 146 GB
- 2 interfaces Gigabit

Software

Compiladores

- gcc 4.1.2 (gfortran, g77, gcc, g++)
- Intel (ifort, icc) v11.081
- Python-1.4.3-24

Librerías matemáticas

- BLACS
- CBLAS (BLAS Library for C)
- FBLAS (BLAS Library for FORTRAN)
- FFTW
- Intel mkl
- LAPACK
- ScaLAPACK

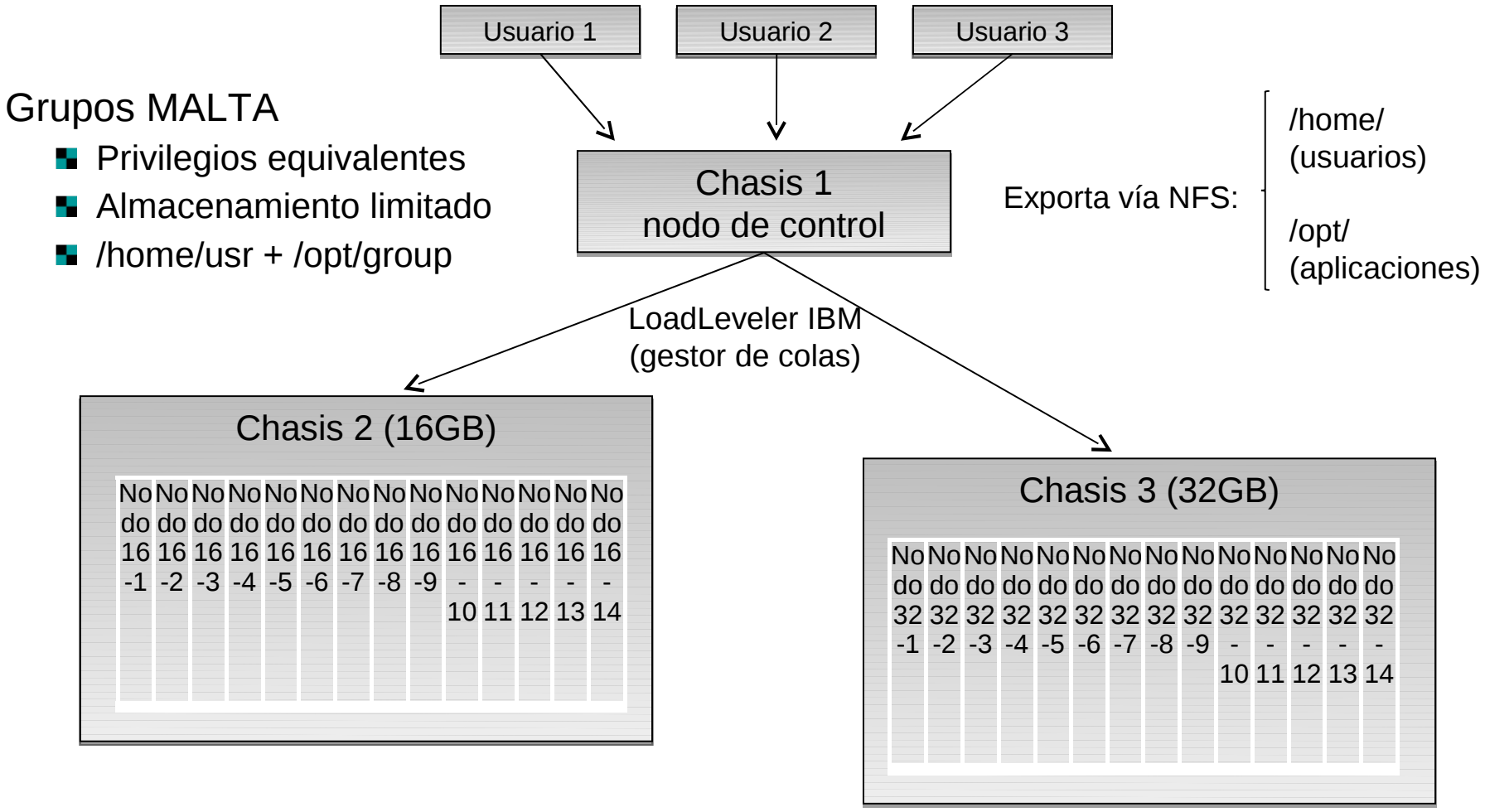
Librerías de cálculo paraleo

- MPICH
- MPICH2
- OpenMPI

Programas

- abinit
- gnuplot
- octave
- siesta
- VASP
- Wien2K
- Crystal06
- Gamess
- Quantum Espresso

Organización



utilizan directorio “nodoXY-Z/scratch”

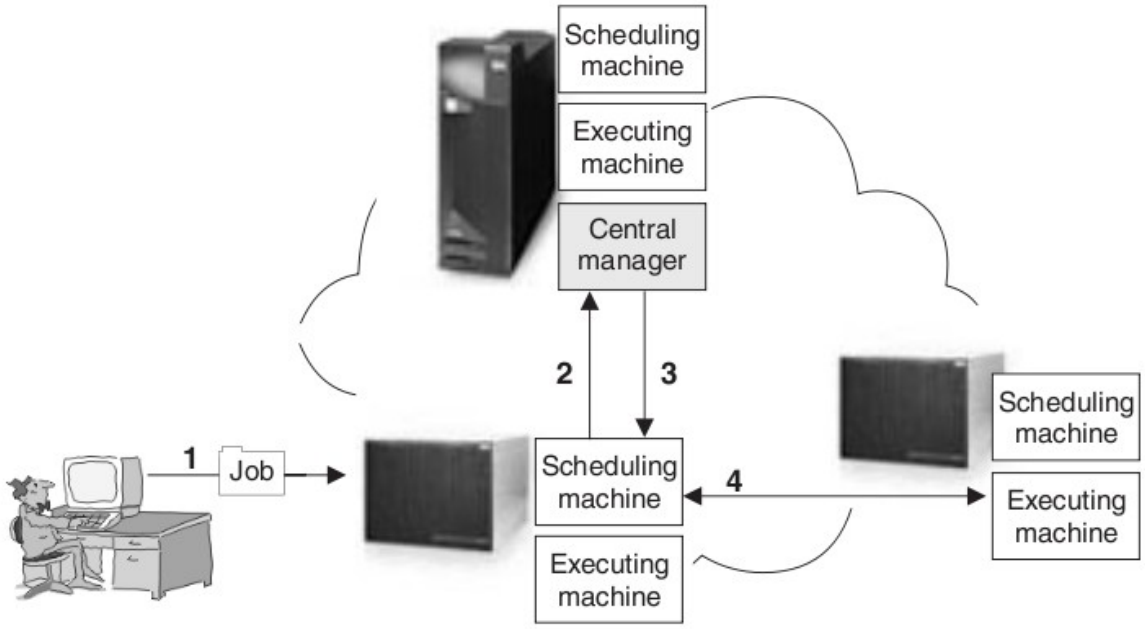
¿Qué es LoadLeveler?

Se trata de un software de planificación que combina las necesidades de proceso y la prioridad de cada trabajo con los recursos disponibles.

Tipos de máquinas:

- Planificador - recibe el trabajo
- Central Manager(CM) – busca una o más máquinas en el cluster que puedan ejecutarlo.
- Executing machine(EM) – es la máquina donde se ejecuta el trabajo
- Submitting machine – envían, cancelan o piden información sobre los procesos en ejecución

¿Cómo planifica trabajos LoadLeveler?



1.- el usuario envía un trabajo que es recibido por el planificador

2.- el planificador contacta con el CM para pedirle que le busque una máquina donde enviar el trabajo

3.- CM analiza las necesidades del trabajo, explora las máquinas disponibles y comunica al planificador cual será la EM (no FCFS)

4.- por último el planificador contacta con EM y le da la información pertinente para ejecutar el proceso (EM y planificador pueden ser el mismo)

Comunicación con LoadLeveler: variables y comandos

La comunicación con Loadleveler se hace a través de un archivo de comandos

Variables de trabajo: útiles cuando se utiliza el fichero .cmd como shell

$(home), $(host), $(jobid), $(user), etc.

LoadL genera una gran cantidad de variables de entorno por defecto:

$LOADL_JOB_NAME, $LOADL_PID, $LOADL_PROCESSOR_LIST, etc.

Variables de entorno generadas por el usuario:

1.- $SCRATCH = es el directorio local donde se de debe ejecutar el trabajo. En él se ha de escribir todo lo relativo a ficheros temporales producidos por el trabajo, así como el output. Este directorio se **borrará automáticamente una vez el trabajo ha** finalizado.

2.- $LL_WORKDIR = directorio de trabajo donde se copiara la salida de LoadLeveler así como el fichero .log creado durante la ejecución.

3.- $LOG = fichero .log creado durante la ejecución del trabajo.

¿Cómo se construye el archivo de comandos?

Trabajos en serie

Hay cuatro clases (=colas) para los trabajos en serie:

1.- sexpress: hasta 2h de cpu
2.- ssmall: hasta 24h de cpu
3.- smedium: hasta 1 semana
4.- slarge: hasta 3 semanas

Ejemplo 1: Cálculo en serie no dependiente

```
#@ job_name = test1                                  ## nombre del trabajo
#@ class = sexpress | ssmall | smedium | slarge     ## cola de trabajo
#@ initialdir = /home/user/test                      ## directorio de trabajo
#@ input = myprog.input                              ## stdin = a.out < myprog.input
#@ output = myprog.output                            ## stdout= a.out >
myprog.out
#@ error = myprog.error                              ## stdout= a.out > myprog.out
#@ executable = myprogram                            ## nombre del ejecuta
#@ arguments = arg1 arg2 arg3                        ## argumentos del ejecutable
#@ queue                                             ## enviar a la cola
```

Ejemplo 2: Cálculo en serie dependiente

```
#  Este fichero comparte el mismo ejecutable
#  en los dos pasos
#
#@ step_name = step1
#@ class = ssmall
#@ executable = executable1
#@ input = step1.in1
#@ output = step1.out1
#@ error = step1.err1
#@ queue
#@ dependency = (step1 == 0)
#@ step_name = ltest
#@ input = longjob.in.$(stepid)
#@ output = longjob.out.$(jobid).$(stepid)
#@ error = longjob.err.$(jobid).$(stepid)
#@ queue
#@ queue
#@ queue
#@ queue
#@ queue
```

Job step	input	output	error
ltest.26.0	longjob.in.0	longjob.out.0	longjob.err.0
ltest.26.1	longjob.in.1	longjob.out.1	longjob.err.1
ltest.26.2	longjob.in.2	longjob.out.2	longjob.err.2
ltest.26.3	longjob.in.3	longjob.out.3	longjob.err.3
ltest.26.4	longjob.in.4	longjob.out.4	longjob.err.4

¿Cómo se construye el archivo de comandos?

Trabajos en paralelo

Hay tres clases (=colas) para los trabajos en serie:

1.- psmall: hasta 24h de cpu – 1 nodo – 8 procesadores
2.- pmedium: hasta 1 semana – 1 nodo – 8 procesadores
3.- plarge: hasta 3 semanas – 1 nodo – 8 procesadores

Ejemplo 3: Cálculo paralelo con OpenMP

```
#@ class = psmall
#@ job_type = parallel
#@ node = 1
#@ tasks_per_node = 6
#@ initialdir = /home/user/myprogs
#@ executable = myopenmpcode
#@ input = inputfile1
#@ output = $(job_name).$(job_step).output
#@ job_cpu_limit = 12:00:00
#@ wall_clock_limit = 20:00:00
#@ error = $(job_name).$(job_step).error
#@ environment = COPY_ALL; OMP_NUM_THREADS=6
#@ queue
```

Ejemplo 4: Cálculo paralelo con MPICH2

```
#!/bin/sh
#
#@ job_name   = sample_mpich
#@ step_name  = step1
#@ job_type   = mpich
#@ output     = test.$(job_name).$(job_step).$(Process).out
#@ error      = test.$(job_name).$(job_step).$(Process).err
#@ class      = pmedium
#@ resources  = ConsumableMemory(1024)ConsumableScratch(10)  ## por
proceso
#@ environment = COPY_ALL
#@ node       = 1                                              ## numero de nodos
#@ tasks_per_node = 1,8                                        ## entre 1-8 procesos
#@ queue
# Copia los datos situados en el directorio
# desde el que se envio el trabajo al directorio
# de trabajo temporal
cp -rp $LL_WORKDIR/data $SCRATCH/
# Se procesan los datos en el $SCRATCH
cd $SCRATCH/data /opt/mpich2/1.0.8/bin/mpirun -
np $LOADL_TOTAL_TASKS  /home/user/prog.exe #Se finalmente transfieren los datos utiles
rm -f $SCRATCH/data/file1
rm -f $SCRATCH/data/file2
rm -f $SCRATCH/data/file3
cp -rp $SCRATCH/data $LL_WORKDIR/
```

Comandos LoadLeveler:

- llsubmit trabajo.cmd: envía el trabajo a la cola

```
[mamel@malta env_variables]$ llsubmit ll_env_var.cmd
llsubmit: The job "malta.84256" has been submitted.
[mamel@malta env_variables]$
```

- llq: muestra el estado de la cola

```
[mamel@malta env_variables]$ llq
Id                       Owner      Submitted   ST PRI Class        Running On
------------------------ ---------- ----------- -- --- ------------ -----------
malta.82126.0            miriam     12/7  14:25 R  50  pmedium      nodo32-3
malta.82128.0            miriam     12/7  14:31 R  50  pmedium      nodo32-4
malta.83217.0            miriam     12/13 00:10 R  50  pmedium      nodo16-3
malta.83546.0            amunoz     12/14 15:16 R  50  pmedium      nodo32-2
malta.84199.0            ruth       12/17 14:19 R  50  ssmall       nodo32-9
malta.84221.0            amorales   12/17 16:49 R  50  ssmall       nodo16-13
malta.84227.0            julia      12/17 17:18 R  50  slarge       nodo16-13
malta.84231.0            placida    12/17 17:28 R  50  psmall       nodo16-4
malta.84232.0            placida    12/17 17:29 R  50  psmall       nodo16-5
malta.84233.0            julia      12/17 17:29 R  50  slarge       nodo16-9
malta.84234.0            julia      12/17 17:30 R  50  slarge       nodo32-9
malta.84238.0            ppc        12/17 17:48 R  50  sexpress     nodo16-11
malta.84239.0            julia      12/17 17:49 R  50  slarge       nodo16-10
malta.84255.0            ppc        12/17 18:46 R  50  sexpress     nodo16-14
malta.84256.0            mamel      12/17 18:50 I  50  pxlarge

31 job step(s) in queue, 1 waiting, 0 pending, 30 running, 0 held, 0 preempted
[mamel@malta env_variables]$
```

Comandos LoadLeveler:

- llq -s trabajo.xyz: muestra información sobre el estado de trabajo.xyz

```
   -----------

      Executable   : /home/mamel/ll_test/env_variables/ll_env_var.cmd
      Exec Args    :
      Num Task Inst: 0
      Task Instance:

   Task
   ----

      Num Task Inst: 0
      Task Instance:

=================== EVALUATIONS FOR JOB STEP malta.84253.0 ====================

The class of this job step is "pxlarge".
Total number of configured initiators of this class on all machines in the cluster: 0
Minimum number of initiators of this class required by job step: 4
The number of configured initiators of this class is not sufficient for this job step.
Not enough resources to start now.
Not enough machines to start step malta.84253.0, machines with available class initiators = 0, needed = 1
Not enough resources for this step as top-dog.
Not enough machines to start step malta.84253.0, machines with available class initiators = 0, needed = 1
[mamel@malta env_variables]$
```

- llcancel trabajo.xyz: cancela el trabajo.xyz

```
[mamel@malta env_variables]$ llcancel malta.84253.0
llcancel: Cancel command has been sent to the central manager.
[mamel@malta env_variables]$
```

Comandos LoadLeveler:

- llclass: muestra las clases (=colas) existentes y grado de ocupación.

  Max Slots: máximo grado de ocupación para cada clase

  Free Slots: número de procesos disponible para cada clase

```
[mamel@malta ~]$ llclass
Name                 MaxJobCPU     MaxProcCPU  Free   Max Description
                    d+hh:mm:ss     d+hh:mm:ss Slots Slots
-------------- -------------- -------------- ----- ----- --------------------
No_Class             undefined      undefined     0     0
sexpress              02:00:00      undefined    41    56
psmall               1+00:00:00      undefined    80   152
ssmall               1+00:00:00      undefined    43    56
pmedium              7+00:00:00      undefined    80   152
smedium              7+00:00:00      undefined    44    56
plarge              19+18:00:00      undefined    80   152
slarge              19+18:00:00      undefined    39    56
------------------------------------------------------------------------------
"Maximum Slots" value of the class "No_Class" is constrained by the MAX_STARTERS
 limit(s).
"Free Slots" values of the classes "No_Class", "sexpress", "psmall", "ssmall", "
pmedium", "smedium", "plarge", "slarge" are constrained by the MAX_STARTERS limi
t(s).
[mamel@malta ~]$
```

Comandos LoadLeveler:

- llstatus: muestra el estado los nodos en el cluster.

  Down: no disponible

  Avail: disponible

  Run: acepta trabajos

  Idle: reposo

  Busy: no acepta trabajos

```
[mamel@malta ~]$ llstatus
Name                 Schedd InQ  Act Startd Run LdAvg Idle Arch      OpSys
malta                Avail   31   31 None     0 0.60     2 Intel64   Linux2
nodo16-10            Avail    0    0 Run      2 2.00  9999 Intel64   Linux2
nodo16-11            Avail    0    0 Run      2 1.92  9999 Intel64   Linux2
nodo16-12            Avail    0    0 Run      3 2.97  9999 Intel64   Linux2
nodo16-13            Avail    0    0 Run      2 2.00  9999 Intel64   Linux2
nodo16-14            Avail    0    0 Run      2 2.98  9999 Intel64   Linux2
nodo16-2             Down     0    0 Down     0 1.07  9999 Intel64   Linux2
nodo16-3             Avail    0    0 Busy     8 8.04  9999 Intel64   Linux2
nodo16-4             Avail    0    0 Busy     8 7.99  9999 Intel64   Linux2
nodo16-5             Avail    0    0 Busy     8 8.00  9999 Intel64   Linux2
nodo16-6             Avail    0    0 Busy     8 8.00  9999 Intel64   Linux2
nodo16-7             Avail    0    0 Busy     8 7.99    30 Intel64   Linux2
nodo16-8             Avail    0    0 Run      3 2.98  9999 Intel64   Linux2
nodo16-9             Avail    0    0 Run      2 1.99  9999 Intel64   Linux2
nodo32-1             Avail    0    0 Idle     0 0.00  9999 Intel64   Linux2
nodo32-10            Avail    0    0 Idle     0 0.00  9999 Intel64   Linux2
nodo32-11            Avail    0    0 Idle     0 0.00  9999 Intel64   Linux2
nodo32-12            Avail    0    0 Busy     8 8.10  9999 Intel64   Linux2
nodo32-13            Avail    0    0 Busy     8 8.00  9999 Intel64   Linux2
nodo32-14            Avail    0    0 Idle     0 0.00  9999 Intel64   Linux2
nodo32-2             Avail    0    0 Busy     8 8.00  9999 Intel64   Linux2
nodo32-3             Avail    0    0 Busy     8 8.03  9999 Intel64   Linux2
nodo32-4             Avail    0    0 Busy     8 8.01  9999 Intel64   Linux2
nodo32-5             Avail    0    0 Idle     0 0.00  9999 Intel64   Linux2
nodo32-6             Avail    0    0 Idle     0 0.00  9999 Intel64   Linux2
nodo32-7             Avail    0    0 Idle     0 0.00  9999 Intel64   Linux2
nodo32-8             Avail    0    0 Run      2 2.00  9999 Intel64   Linux2
nodo32-9             Avail    0    0 Run      2 2.05  9999 Intel64   Linux2

Intel64/Linux2          28 machines     31  jobs    100  running tasks
Total Machines          28 machines     31  jobs    100  running tasks

The Central Manager is defined on malta

The BACKFILL scheduler is in use

All machines on the machine_list are present.
[mamel@malta ~]$
```

Futuros trabajos:

- Instalación de nuevo software (abierto a usuarios)
- Aumentar potencia de cálculo (adquisición de nuevos nodos)
- Sistema de almacenamiento y back up
- Sistema de ficheros global para paralelización

Material de referencia

- Web del proyecto MALTA – Consolider:
  - http://www.malta-consolider.com
- Clúster Information Center (LoadLeveler):
  - http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ib
- Site de Descargas: (Actualizaciones)
  - https://www14.software.ibm.com/webapp/set2/sas/f/loadleveler/home.html