



# DEISA Batch Systems

G. Pringle

(c) 2004 - 2011 DEISA



DEISA is funded by the European Commission in FP7 under grant agreement RI-222919



# Table of contents

1 Introduction.....	1
1.1 Overview .....	1
2 Load Leveler .....	3
2.1 LoadLeveler Commands.....	3
2.2 Remote job management.....	3
2.3 LoadLeveler Directives.....	3
2.3.1 LoadLeveler General Directives .....	4
2.3.2 LoadLeveler “POWER” Directives .....	4
2.3.3 LoadLeveler “BlueGene” Directives .....	6
3 PBS Pro .....	9
3.1 PBS Pro Commands.....	9
3.2 PBS Pro Directives .....	9
3.3 Example MPI script .....	10
3.4 CSC: Louhi .....	10
3.5 EPCC: HECToR XE6.....	10
3.5.1 Local executable statements .....	10
3.6 LRZ: HLRB II.....	11
3.6.1 Local PBS Pro Directives .....	11
3.6.2 Local executable statements .....	11
4 LSF.....	13
4.1 LSF Commands .....	13
4.2 LSF Directives .....	13
5 NQS II.....	15
5.1 NQS II Commands.....	15
5.2 NQS II Directives .....	15
5.3 HLRS: SX_9 specifics .....	16
5.3.1 Local NQS II Directives .....	16
5.3.2 Example MPI script .....	16
6 Moab+Slurm .....	19
6.1 Moab+slurm Commands.....	19
6.2 Moab Directives.....	19
6.3 BSC: MN specifics .....	20
6.3.1 Example MPI script .....	20



# 1 Introduction

This document briefly describes each of the Batch Systems currently deployed at each of the DEISA platforms. It describes some basic commands of each system and then lists some site-specific commands which may be of interest. For further more detailed information, the user should visit the platform's online User Guide.

## 1.1 Overview

The following table holds a list of the current DEISA platforms, their associated DEISA site, and the batch system employed on the platform.

<b>DEISA Site: Platform</b>	<b>Batch System</b>
BSC: MareNostrum (IBM PowerPC)	Moab+Slurm
CINECA: CNE-SP6 (IBM SP6)	LoadLeveler
CSC: Louhi (Cray XT4/XT5)	PBS Pro
EPCC: HECToR (Cray XE6)	PBS Pro
FZJ: JUGENE (IBM BG/P)	LoadLeveler
HLRS: SX-9 (NEC SX-9)	NQS II
IDRIS: Vargas (IBM Power6)	LoadLeveler
IDRIS: Babel (IBM BG/P)	LoadLeveler
LRZ: HLRB II (SGI Altix 4700)	PBS Pro
RZG: VIP (IBM Power6)	LoadLeveler
RZG: Genius (IBM BG/P)	LoadLeveler
SARA: Huygens (IBM Power6)	LoadLeveler

*Table 1: DEISA platforms*



## 2 Load Leveler

This Section describes common LoadLeveler batch commands, LoadLeveler directives and, finally, an example MPI batch script.

### 2.1 LoadLeveler Commands

```
llsubmit <job_script>
```

Submit a job script, called 'job\_script', for execution

```
llq
```

Check the status of your job(s).

```
llcancel <job_id>
```

Cancel a job.

```
llstatus
```

Returns status of machine.

### 2.2 Remote job management

If your Home Site is either CINECA, IDRIS, FZJ or RZG and your Execution Platforms include CINECA CNE-SP6, IDRIS Vargas and/or RZG VIP, then you can manage your remote jobs directly from your Home Site using the `-X` flag. For instance, if you wish to submit your job to a remote site, use

```
llsubmit -X <site>
```

where `<site>` is one of the following: `deisa_cne`, `deisa_idr`, `deisa_fzj` or `deisa_rzg`.

The `-X` flag can be used with all the LoadLeveler commands listed above, i.e. if you wish to check the status of your remote job, use

```
llq -X <site>
```

etc.

### 2.3 LoadLeveler Directives

This section lists some typical and important LoadLeveler directives accepted or unadvised. A general overview of LoadLeveler is provided by the corresponding IBM documentation[1].

## LoadLeveler Directives

### 2.3.1 LoadLeveler General Directives

```
#@ wall_clock_limit = <HH:MM:SS> [,HH:MM:SS]
```

The first time sets the hard limit of the execution time in hours, minutes and seconds. The second optional time sets a soft limit whereby the process receives a warning at this time.

```
#@ requirements = (Feature == "DEISA")
```

This directive is required for all DEISA jobs, and must be employed.

```
#@ notify_user = <your-personal-email>
```

Emails the user once the job is complete. Note: Use your private email and not an account local to the execution platform.

```
#@ notification = error
```

Notifies the job only when the job fails.

```
#@ shell = /bin/bash
```

Specifies which shell to employ.

```
#@ queue
```

Must appear at the end of the list of directives.

Please be aware that using some other keywords can on the contrary produce unexpected behavior. Therefore, the following keywords must be avoided in DEISA jobs:

`#@ requirements = (X)` with a value for X different from `Feature == "DEISA"`

`#@ class`

### 2.3.2 LoadLeveler "POWER" Directives

```
#@ total_tasks = <number of cores>
```

Sets number of cores for an MPI job.

```
#@ resources = ConsumableCpus(N)
```

For multithread parallel jobs.

```
#@ job_type = parallel
```

Otherwise, could be set to serial if needed.

```
#@ data_limit = MEMORY
```

- 
1. [http://publib.boulder.ibm.com/infocenter/cresctr/vrx/topic/com.ibm.cluster.loadl35.admin.doc/am2ugmst\\_xtoc.html](http://publib.boulder.ibm.com/infocenter/cresctr/vrx/topic/com.ibm.cluster.loadl35.admin.doc/am2ugmst_xtoc.html)

## LoadLeveler Directives

where MEMORY is the amount of data memory required. This directive is optional: the default value is set to the recommended value.

```
#@ stack_limit = MEMORY
```

same as data\_limit above, but for stack memory. This directive is optional: the default value is set to the recommended value.

Please be aware that using some other keywords can on the contrary produce unexpected behavior. Therefore, the following keywords must be avoided in DEISA jobs:

#@ blockingwith a value different from unlimited

#@ max\_processors

#@ min\_processors

#@ network.X=Y

#@ node

#@ node\_usage

#@ task\_geometry

#@ tasks\_per\_node

Example of a POWER MPI script for DEISA users

```
#@ job_name = MyJob
#@ output = MyModel/output.$(jobid).log_ll
#@ error = MyModel/output.$(jobid).log_ll
#@ notify_user = <replace with your email>
#@ notification = error
#@ shell = /bin/bash
#@ requirements = (Feature == "DEISA")
#@ job_type = parallel
#@ total_tasks = 128
#@ wall_clock_limit = 06:00:00,05:50:00
#@ data_limit = 512mb
#@ stack_limit = 400mb
#@ queue

module load deisa cpmd
cd $DEISA_SCRATCH
cp $DEISA_DATA/MyModel/CPMD/* .
$CPMD input.in > cpmd.out
cp * $DEISA_DATA/MyModel/CPMD
```

Note that, in this example, both the standard output and standard error are sent into the same file MyModel/output.\$(jobid).log\_ll which will reside in the \$DEISA\_HOME directory.

## 2.3.3 LoadLeveler “BlueGene” Directives

```
#@ job_type = BLUEGENE
```

is required to setup a step running on the BG/P.

```
#@ bg_size = xxxxx
```

is required to design the size of a BG job

```
#@ bg_connection = {MESH|TORUS|PREFER_TORUS}
```

Multisteps jobs are accepted (with steps running on the Front-end Node for pre- and post-processing purposes).

You must also be aware that using some other keywords can on the contrary produce unexpected behavior. Therefore, the following keywords must be avoided in DEISA jobs:

```
#@ bg_shape
```

```
#@ bg_partition
```

Example of BlueGene MPI script for DEISA users

```
#!/bin/bash
# @ job_name = myjob
# @ error = $(job_name).$(jobid).out
# @ output = $(job_name).$(jobid).out
# @ wall_clock_limit = 00:30:00
# @ notify_user = <replace with your email>
# @ notification = error
# @ job_type = bluegene
# @ bg_size = 128
# @ queue
#
# Run the program in Virtual Node Mode on the BlueGene/P:
#
# Executable statements follow
```

Executable statements for BlueGene DEISA users

```
module load deisa
cp my_code $DEISA_SCRATCH
# Warning: if you need to transfer important volumes
# of data, please use a multi-step job

cp input.data $DEISA_SCRATCH
cd $DEISA_SCRATCH

mpirun -mode VN -np 256 -mapfile TXYZ -exe ./my_code
```

## LoadLeveler Directives

```
# $LOADL_STEP_INITDIR is the submission directory  
cp output.data $LOADL_STEP_INITDIR
```

Please be aware that \$DEISA\_DATA and \$DEISA\_HOME are not usable under BlueGene job step type, and using some can produce unexpected behavior.



## 3 PBS Pro

### 3.1 PBS Pro Commands

```
qsub <jobscript>
```

Submit a jobscript for execution.

```
qstat
```

Display all queued jobs

```
qstat -Q
```

Status of queues

```
qdel <job_id>
```

Delete a job

### 3.2 PBS Pro Directives

```
#PBS -l mppwidth= <number of nodes>
```

Sets the number of nodes

```
#PBS -l mppnppn = <number of cores per node>
```

Sets the number of cores per node

```
#PBS -l walltime = <HH:MM:SS>
```

Sets the maximum wall clock time for the job

```
#PBS -j oe
```

Merges both standard output and standard error

```
#PBS -m e
```

Send email when job is done

```
#PBS -M user@home.com
```

Email address for notification email.

### 3.3 Example MPI script

```
#!/bin/bash

#PBS -N test
#PBS -j oe
#PBS -l walltime=1:00:00
#PBS -l mppwidth=256
#PBS -m abe
#PBS -M user@home.com
#PBS -r n

module load deisa
cp input.dat
$DEISA_SCRATCH
cp program $DEISA_SCRATCH

cd $DEISA_SCRATCH
aprun -n 256 ./program
cp output.dat ~/.
```

### 3.4 CSC: Louhi

For more information on the Cray XT4/XT5 at CSC, please visit: [http://www.csc.fi/english/pages/louhi\\_guide/batch\\_jobs/parallel\\_jobs](http://www.csc.fi/english/pages/louhi_guide/batch_jobs/parallel_jobs)

### 3.5 EPCC: HECToR XE6

This section describes some localised PBS Pro directives particular to the Cray XE6 at EPCC. For more information, please visit <http://www.hector.ac.uk/support/documentation/userguide/batch.php>

#### 3.5.1 Local executable statements

```
export TASKS=`qstat -f $PBS_JOBID | awk '/mppwidth/ {print $3}`
export TASKSPERNODE=`qstat -f $PBS_JOBID | awk '/mppnppn/ {print $3}`
aprun -n $TASKS -N $TASKSPERNODE ./program
```

## 3.6 LRZ: HLRB II

This section describes some localised PBS Pro Directives particular to the SGI Altix at LRZ. For more information, please visit: <http://www.lrz-muenchen.de/services/compute/hlrb/batch/batch.html>

### 3.6.1 Local PBS Pro Directives

The PBS directives `mppwidth` and `mppnppn` are not available on HLRB II. Their function is subsumed by the `select` directive, as illustrated in the following example.

```
#PBS -l select=280:ncpus=1
```

### 3.6.2 Local executable statements

```
. /etc/profile.d/modules.sh
module load deisa
cd $DEISA_DATA
mpiexec ./myprog
```



## 4 LSF

### 4.1 LSF Commands

**bqueues -l**

Display queue information

**bsub < <jobscript>**

Submits the commands contained in <jobscript>. NB: the input jobscript MUST be redirected with the "<" character.

**bjobs -a**

Shows submitted jobs

**bkill <jobid>**

Cancels the job <jobid> from the queuing system.

**bhist <jobid>**

Display historical information about the job <jobid>

### 4.2 LSF Directives

**#BSUB -n <number of cores>**

Specifies number of cores

**#BSUB -W <HH:MM>**

Sets maximum wall clock time

**#BSUB -o %J.out**

Sets location of standard output

**#BSUB -e %J.err**

Sets location of standard error



# 5 NQS II

## 5.1 NQS II Commands

```
qsub <jobscript>
```

Submit the jobscript file

```
qstat
```

Display all queued jobs

```
qstat -Q
```

Status of queues

```
qdel <job_id>
```

Delete a job

## 5.2 NQS II Directives

```
#PBS -q multi
```

queue: dq for <=8 CPUs

```
#PBS -T mpisx
```

Job type: mpisx for MPI

```
#PBS -l cpunum_job= <number of cores per node>
```

cpus per Node

```
#PBS -b <number of nodes>
```

number of nodes

```
#PBS -l elapstim_req= <HH:MM:SS>
```

max wallclock time

```
#PBS -l memsz_job=<memory per node>
```

memory per node

```
#PBS -A <account code>
```

Your Account code

## HLRS: SX\_9 specifics

```
#PBS -N MyJob
```

job name

```
#PBS -M user@home.com
```

you should always specify your email address.

### 5.3 HLRS: SX\_9 specifics

This section describes some NQS II Directives particular to the SX-9 at HLRS. For more information, please visit the related documentation at [HLRS\[1\]](#).

#### 5.3.1 Local NQS II Directives

There are no localised differences to NQS II on the SX-9.

#### 5.3.2 Example MPI script

```
#!/usr/local/bin/bash
#PBS -q multi
#PBS -T mpisx
#PBS -l cpunum_job=8
#PBS -b 4
#PBS -l elapstim_req=02:00:00
#PBS -l cputim_job=08:00:00
#PBS -l cputim_prc=07:55:00
#PBS -l memsz_job=10gb
#PBS -A <account code>
#PBS -j o
#PBS -N MyJob
#PBS -M jo@user

SCR='ws_allocate Run1 2'
cd $SCR

export OMP_NUM_THREADS=8
export MPIPROGINF=YES
export F_FILEINF=YES
export MPIMULTITASKMIX=YES
MPIEXPORT="OMP_NUM_THREADS F_FILEINF"
```

---

1. [https://wickie.hlr.de/platforms/index.php/Batch\\_system](https://wickie.hlr.de/platforms/index.php/Batch_system)

## HLRS: SX\_9 specifics

```
export MPIEXPORT  
  
module load deisa  
mpirun -nn 4 -nnp 1  
./mycode  
cp outfile $HOME/code
```



## 6 Moab+Slurm

This section lists some typical and important Moab+Slurm directives accepted or unadvised. For a general overview of Moab+Slurm, please visit <http://www.clusterresources.com/products/mwm/docs/index.shtml>

### 6.1 Moab+slurm Commands

```
mnsubmit <jobscript>
```

Submit the job script file

```
mnq
```

Display all your queued jobs

```
checkjob <job_id>
```

Verbose display of job <job\_id>

```
mncancel <job_id>
```

Delete a job

### 6.2 Moab Directives

#### Moab Directives

```
# @ job_name = <name of the job>
```

Sets the name of the job.

```
# @ output = <output file>
```

Sets the file where stdout will be fetched.

```
# @ error = <error file>
```

Idem than output but with stderr.

```
# @ total_tasks = <number of tasks>
```

Sets number of tasks.

```
# @ tasks_per_node = <number of cores>
```

Sets number of tasks within each node.

```
# @ wall_clock_limit = <HH:MM:SS>
```

Sets the maximum execution time in hours, minutes and seconds

### 6.3 BSC: MN specifics

This section describes some Moab+Slurm Directives particular to MareNostrum at BSC. For more information, please visit the corresponding documentation at BSC[1].

#### 6.3.1 Example MPI script

```
# @ job_name = MyJob
# @ initialdir = .
# @ output = MyModel/output.%j.out
# @ error = MyModel/output.%j.err
# @ total_tasks = 128
# @ cpus_per_task = 4
# @ tasks_per_node = 1
# @ wall_clock_time = 06:00:00

module load deisa cpmd
cd $DEISA_SCRATCH
cp $DEISA_HOME/MyModel/CPMD/* .
srun $CPMD input.in > cpmd.out
cp * $DEISA_HOME/MyModel/CPMD
```

---

1. [http://www.bsc.es/plantillaA.php?cat\\_id=596](http://www.bsc.es/plantillaA.php?cat_id=596)