Grid User Guide

Introduction

This user's guide is intended to provide the minimum amount of information needed by a new user of this system. As such, it assumes that the user is familiar with basic notions on scientific computing, in particular the basic commands of the Unix operating system.

The information in this guide includes basic technical documentation about the the software environment, and also on available applications.

Please read it carefully and if any doubt arises don't hesitate to contact our support mail (see below in Support).

Usage

The $HOME directories are shared between the UIs and the computing nodes. There is a *projects* shared area (located at /gpfs/projects/), also accessible from the UI and the computing nodes. If your group does not have this area, please open an Incidence ticket.

The shared directories **are not intended for scratch**, use the temporal areas of the local filesystems instead. In other words, instruct every job you send to copy the input from the shared directory to the local scratch ($TMPDIR), execute all operations there, then copy the output back to some shared area where you will be able to retrieve it comfortably from the UI.

As mentioned above, the contents of $TMPDIR are removed after job execution.

Disk quotas

Disk quotas are enabled on both user and projects filesystems. A message with this information should be shown upon login. If you need more quota on your user space (not in the project shared area), please contact the system administrators explaining your reasons.

If you wish to check your quota at a later time, you can use the commands `mmlsquota gpfs_csic` (for user quotas) and `mmlsquota -g id -g gpfs_projects` (for group quotas).

```
********************************************************************
                   INFORMATION ABOUT YOUR CURRENT DISK USAGE

USER               Used      Soft      Hard     Doubt     Grace
Space (GB):        3.41     50.00      0.00      0.06      none
Files (x1000):       64         0         0         0      none

GROUP              Used      Soft      Hard     Doubt     Grace
Space (GB):        0.00   1000.00     00.00      0.00      none
Files (x1000):        0         0         0         0      none
********************************************************************
```

For a basic interpretation of this output, note that the "Used" column will tell you about how much disk space you are using, whereas "Soft" denotes the limit this "Used" amount should not exceed. The "Hard" column is the value of the limit "Used" plus "Doubt" should not cross. A healthy disk space management would require that you periodically delete unused files in your $HOME directory, keeping its usage below the limits at all times. In the event that the user exceeds a limit, a grace period will be shown in the "Grace" column. If the user does not correct the situation within the grace period, she will be banned from writing to the disk.

For further information you can read the mmlsquota command manual page.

Connecting to Grid cluster

This cluster is comprised of a pool of machines reachable through a single entry point. The connections to the internal machines are managed by a director node that tries to ensure that proper balancing is made across the available nodes at a given moment.

Please note that this cluster is *not intended for the execution of CPU intensive tasks*, for this purpose use any of the available computing resources. Every process spawned is limited to a maximum CPU time of 2 hours.

Login on these machines is provided via Secure Shell. Outgoing SSH connections are not allowed by default from this cluster. Inactive SSH sessions may be closed after 12h. It is highly recommended that you set up SSH Keys for authentication, instead of using your username and password.

Hostname	Operating System	SSH server key fingerprint
gridui.ifca.es	CentOs 7	SHA256:iuRWTlWP7db1cM0d58m5TyGzPnZoDCNoBM3RuTtxddU

Batch System

The Batch System is based on CentOs 7 machines, running on x86_64.

Local submission is allowed for certain projects. As stated below, there are some shared areas that can be accessed from the computing nodes. The underlying batch system is Slurm.

SLURM Concept

SLURM manages user jobs with the followingkey characteristics:

- set ofrequested resources:
  - number of computing resources: nodes(including all their CPUsand cores) or CPUs(including all their cores) or cores
  - amount ofmemory: either per node or per (logical) CPU
  - the (wall)time needed for the user's tasks to complete their work
- a set ofconstraints limiting jobs to nodes with specific features
- a requested node partition(job queue)
- a requested quality of service(QoS) level which grants users specific accesses
- a requested account for accounting purposes

Running Jobs

As is defined above SLURM is the utility used for batch processing support, so all jobs must be run through it. This part provides information for getting started with job execution (see the official slurm documentation to get more information on how to create a job) .

NOTE: In order to keep the login nodes in a proper load, a 10 minutes limitation in the CPU time is set for processes running interactively in these nodes. Any execution taking more than this limit should be carried out through the queue system.

Manage Jobs

A job is the execution unit for the SLURM. A job is defined by a script containing a set of directives describing the job, and the commands to execute.

These are the basic directives to submit jobs:

- **sbatch <job script>:** submits a *job script* to the queue system (see below for job script directives).
- **squeue [-u user]:** shows all the jobs submitted by all users or by the user if you specify the **-u** option.
- **scancel <job id>:** removes his/her job from the queue system, canceling the execution of the job if it was already running.
- **srun:** submit interactive job, run (parallel) job step
- **squeue:** view queued jobs
- **sinfo:** view partition and node info
- **scontrol:** detailed control and info on jobs, queues, partitions
- **sstat:** view system-level utilization (memory, I/O, energy)for running jobs / job steps
- **sacct:** view system-level utilizationfor completed jobs / job steps (accounting DB)
- **sprio:** view job priority factors
- **sshare:** view accounting share info (usage, fair-share, etc.)

Job directives

A job must contain a series of directives to inform the batch system about the characteristics of the job and you can configure them to fit your needs. These directives appear as comments in the job script and usually at the top just after the shebang line, with the following syntax:

```
#SBATCH option=value
```

Note that these directives:

- start with the #SBATCH prefix
- are always lowercase
- have no spaces in between.
- don't expand shell variables (they are just shell comments)

This table describes the common directives you can define in your job (see below an example):

`--job-name=value`	The name of the job that appears in the batch queue	*script_name*
`--partition=...`	The name of the queue in slurm	*compute*
`--output=...`	The name of the file to collect the standard output of the job. The %j part in the job directives will be sustitute by the job ID.	*file-%j.out*
`--error=...`	The name of the file to collect the standard error of the job. The %j part in the job directives will be sustitute by the job ID.	*file-%j.err*
`--chdir=...`	The working directory of your job (i.e. where the job will run). If not specified, it is the current working directory at the time the job was submitted.	submitting directory
`--qos=...`	Quality of Service (or queue) where the job is allocated. By default, a queue is assigned for the user so this variable is not mandatory.	main
`--time=...`	The limit of wall clock time. This is a mandatory field and you must set it to a value greater than the real execution time for your application and smaller than the time limits granted to the user. Notice that your job will be killed after the elapsed period. The format can be: m, m:s, h:m:s, d-h, d-h:m or d-h:m:s	qos default time limit
`--ntasks=...`	The number of processes to allocate as parallel tasks.	1
`--cpus-per-task=...`	Number of processors for each task. Without this option, the controller will just try to allocate one processor per task. The number of cpus per task must be between 1 and 16, since each node has 16 cores (one for each thread).	1
`--ntasks-per-node`	The number of tasks allocated in each node. When an application uses more than 3.8 GB of memory per process, it is not possible to have 16 processes in the same node and its 64GB of memory. It can be combined with the cpus_per_task to allocate the nodes exclusively, i.e. to allocate 2, processes per node, set both directives to 2. The number of tasks per node must be between 1 and 16.	1
`--mem-per-cpu`	Minimum memory required per allocated CPU. Default units are megabytes unless the SchedulerParameters configuration parameter includes the "default_gbytes" option for gigabytes.	DefMemPerCPU

Job environment variables

There are also a few SLURM environment variables you can use in your scripts:

SLURM_JOBID	Specifies the job ID of the executing job
SLURM_NPROCS	Specifies the total number of processes in the job. Same as **-n, --ntasks**
SLURM_NNODES	Is the actual number of nodes assigned to run your job
SLURM_PROCID	Specifies the MPI rank (or relative process ID) for the current process. The range is from 0-(**SLURM_NPROCS**-1)
SLURM_NODEID	Specifies relative node ID of the current job. The range is from 0-(**SLURM_NNODES**-1)
SLURM_LOCALID	Specifies the node-local task ID for the process within a job
SLURM_NODELIST	Specifies the list of nodes on which the job is actually running
SLURM_SUBMIT_DIR	The directory from which **sbatch** was invoked.
SLURM_MEM_PER_CPU	Memory available per CPU used

Interactive batch jobs

In order to launch an interactive session on a compute node do:

```
srun --time=1:00:00 --ntasks 2 --nodes=1 --account=chpc --partition=ember --pty /bin/tcsh -l
```

The important flags are the `--pty` denoting an interactive terminal, and `/bin/tcsh -l`, which is the shell to run. If you prefer bash, replace `/bin/tcsh` with `/bin/bash`. As in submitting a batch script, the -n specifies tasks and the -N specifies nodes. The -l (or --label) will prepend task number to lines of stdout/err. The --label option will prepend lines of output with the remote task id. Note that the order of the command is important, the "--pty /bin/tcsh -l" has to be at the end.

The srun flags can be abbreviated as:

```
srun -t 1:00:00 -n 2 -N 1 -A chpc -p ember --pty /bin/tcsh -l
```

The srun command by default passes all environment variables of the parent shell therefore the X window connection gets preserved as well, allowing for running graphical applications such as GUI based programs inside the interactive job.

Cluster queues tend to be very busy; it may take some time for an interactive job to start. For this reason, they have added two nodes in a special partition on the notchpeak cluster that are geared more towards interactive work. Job limits on this partition are 8 hours wall time, a maximum ten submitted jobs per user, with a maximum of two running jobs with a maximum total of 32 tasks and 128 GB memory. To access this special partition, called `notchpeak-shared-short`, request both an account and partition under this name, e.g.:

```
srun -N 1 -n 2 -t 2:00:00 -A notchpeak-shared-short -p notchpeak-shared-short --pty /bin/bash -l
```

Job examples

Example for a sequential job:

```
#!/bin/bash
#
#SBATCH --job-name=hello
#SBATCH --output=hello.out
#SBATCH --ntasks=1
#SBATCH --time=10:00
# From here the job starts
echo hostname
sleep 60
```

```
#!/bin/bash
#
#SBATCH --job-name=hello
#SBATCH --output=hello.out
#SBATCH --ntasks=4 # The job spawns in 4 cores
#SBATCH --time=10:00
# From here the job starts
srun parallel.sh
srun sleep 60
```

Example for a parallel job:

Example for an interactive job:

```
srun -p interactive --qos qos-interactive --time=0:30 -N2 --ntasks-per-node=4 --pty bash -isalloc -p
interactive --qos qos-interactive bash -c‘ssh -Y $(scontrol show hostnames | head-n 1)’srun -p interactive --
qos qos-besteffort --cpu-bind=none -N1 -n4 --pty bash -isrun -C skylake -p batch --time=0:10:0 -N1 -c28 --pty
bash -i
```

Example for a batch job:

```
sbatch job.sh
sbatch -N 2 job.sh
sbatch -p batch --qos qos-batch job.sh
sbatch -p long --qos qos-long job.sh
sbatch --begin=2019-11-23T07:30:00 job.sh
sbatch -p batch --qos qos-besteffort job.sh
sbatch -p bigmem --qos qos-bigmem --mem=2T job.sh
```

Example for status and details for partitions, nodes, reservations:

```
squeue / squeue -l / squeue -la / squeue -l -p batch / squeue -t PD
scontrol show nodes / scontrol show nodes $nodename
sinfo / sinfo -s / sinfo -N
sinfo -T
```

Extra utils/Software

If you need some extra software, do not hesitate to Open a ticket.

Support

IFCA provides to users consulting assistance. User support consultants are available during normal business hours, Monday to Friday, 09 a.m. to 17 p.m. (CEST time).

User questions and support are handled at: <grid.support@ifca.unican.es>

If you need assistance, please supply us with the nature of the problem, the date and time that the problem occurred, and the location of any other relevant information, such as output or log files.

Before opening a new incidence, please check the Frequently Asked Questions page

Questions, support and/or feedback should be directed through the use the Helpdesk.