How to use Marenostrum 4

Félix Ramos
Francisco González
Ricard Zarco
Helena Gómez

Basic usage guide
BSC HPC Access

- Access through SSH
  - OpenSSH for Linux / OSX
  - PuTTY for Windows
- On the facilities granted you can authenticate by:
  - password
  - SSH public keys
- But, how to...
  ...Generate & use keys:
  - `$ ssh-keygen -t rsa`
  - `$ ssh-copy-id bsc99999@mn3.bsc.es`
Shared Credentials

- Shared users among:
  - Data Transfer
  - MareNostrum IV
  - Some other HPC machines
- Same $HOME (/gpfs/home/bscXX/bscXXXXX/)
- Same $USER (bscXXXXX) & passwd (centrally managed)
Hands on 0: Change your password
General Parallel Filesystem (GPFS)

- High performance parallel filesystem
  - `/gpfs/apps (Support vetted applications)`
    Python2, Python3, Tensorflow, namd, gromacs...
  - `/gpfs/home (User's home, backup)`
    Scripts, codes, documents...
  - `/gpfs/projects (Inputs, custom installations, backup)`
    Execution folders, init data...
  - `/gpfs/scratch (Temporary files, NO backup)`
    Execution data, Huge log/output files
  - `/gpfs/archive (Long term storage, batch interaction)`
    Data not in usage, long term storage
Filesystem limits (Quota)

- Filesystem limit per user and/or group.
- Check with: bsc_quota

```
@login3:~> bsc_quota
<table>
<thead>
<tr>
<th>Fylesystem</th>
<th>type</th>
<th>blocks</th>
<th>quota</th>
<th>limit</th>
<th>in_doubt</th>
<th>grace</th>
<th>files</th>
<th>in_doubt</th>
</tr>
</thead>
<tbody>
<tr>
<td>gpfs_home</td>
<td>USR</td>
<td>22.43M</td>
<td>40.00G</td>
<td>41.00G</td>
<td>0</td>
<td>none</td>
<td>801</td>
<td>0</td>
</tr>
<tr>
<td>gpfs_projects</td>
<td>GRP</td>
<td>298.01G</td>
<td>9.77T</td>
<td>10.25T</td>
<td>0</td>
<td>none</td>
<td>7129</td>
<td>0</td>
</tr>
<tr>
<td>gpfs_scratch</td>
<td>GRP</td>
<td>469.00M</td>
<td>58.59T</td>
<td>51.52T</td>
<td>0</td>
<td>none</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>gpfs_archive</td>
<td>GRP</td>
<td>0</td>
<td>9.766T</td>
<td>10.25T</td>
<td>0</td>
<td>none</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
```

- Typical related problems:
  - Job submission failure when $HOME is over quota
  - Job execution failure when writing to over-quota filesystem
- Group-SHARED quota on /gpfs/projects and /gpfs/scratch
GPFS Filesystem Performance

Useful tips...

• Stay below your quota
  – Near your limit it slows down

• Keep the file number per directory low
  – Keep below 1000 files per directory

• Avoid creating/writing many files at the same time in the same place
  – Will compromise I/O speed

• Avoid small files (blocksize 16 MB)
  – Meaning that the smaller file will weight at least 16MB
/gpfs/home Filesystem usage

- Few space (~40 GB per user)

- /gpfs/home Do:
  - Store source code
  - Store personal scripts

- /gpfs/home Don't:
  - Use as production directory
/gpfs/archive Filesystem usage

• You can check availability with bsc_quota and dtquota

• /gpfs/archive Do:
  – Store data you are not going to use soon
  – Store processed final results

• /gpfs/archive Don't:
  – Execute commands interactively (cp, mv, ...)
  – Try to put ACLs
Data Transfer Commands

• Set of commands to send data transfer jobs to queues
  - Available in MareNostrum and dt01

• Available commands:

<table>
<thead>
<tr>
<th>File movement</th>
<th>Archiving &amp; synchronizing</th>
<th>Job Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>- dtcp</td>
<td>- dttar</td>
<td>- dtq</td>
</tr>
<tr>
<td>Copying files</td>
<td>Compressing files</td>
<td>Check datajobs status</td>
</tr>
<tr>
<td>- dtmv</td>
<td>- dtrsync</td>
<td>- dtcancel</td>
</tr>
<tr>
<td>Moving files</td>
<td>Synchronizing files</td>
<td>Cancel datajobs</td>
</tr>
</tbody>
</table>

[Attention] must load transfer module to use them in MN4:
$ module load transfer/1.0
Node’s local disk (/scratch)

- All nodes have disk for temporary files
  - Accessible via $TMPDIR
  - Not shared between nodes (different to /gpfs/scratch)
  - Content erased after execution

- Useful for temporary files
  - Temporal data from MPI communication

- 200 GB disk

<table>
<thead>
<tr>
<th>Directory</th>
<th>Size (in bytes)</th>
<th>Free (in bytes)</th>
<th>Total (in bytes)</th>
<th>Used (%)</th>
<th>Directory Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/sda2</td>
<td>226775428</td>
<td>166182416</td>
<td>49050436</td>
<td>78%</td>
<td>/scratch</td>
</tr>
<tr>
<td>gpfs_scratch</td>
<td>547535945728</td>
<td>4092991160320</td>
<td>1382544785408</td>
<td>75%</td>
<td>/gpfs/scratch</td>
</tr>
<tr>
<td>gpfs_apps</td>
<td>8589934592</td>
<td>6371782144</td>
<td>2218152448</td>
<td>75%</td>
<td>/gpfs/apps</td>
</tr>
<tr>
<td>gpfs_home</td>
<td>17179869184</td>
<td>15056143104</td>
<td>2123726080</td>
<td>88%</td>
<td>/gpfs/home</td>
</tr>
<tr>
<td>gpfs_projects</td>
<td>27377711118592</td>
<td>1427638181888</td>
<td>1310132936704</td>
<td>53%</td>
<td>/gpfs/projects</td>
</tr>
</tbody>
</table>
Managing permissions for data access

- Do not alter basic access permissions
  - Unwanted access, accidental alteration/deletion
  - Coarse granularity

- Use ACLs
  - Per user/group access to files and directories
  - Small granularity

- When in doubt, ask Support Team
Hands-on 1: Data movement
MareNostrum logins

- 3 external accessible logins:
  - mn1.bsc.es
  - mn2.bsc.es
  - mn3.bsc.es

- No outgoing connections
  - No downloads or uploads
  - 5 minutes cpu time limit:
    - For long data transfers, use dt nodes
    - For long compilations, use interactive nodes:
      - $ salloc -p interactive

- 2 internal accessible logins:
  - Login4
  - Login5
# Login usage

<table>
<thead>
<tr>
<th>✅</th>
<th>❌</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage &amp; edit files</td>
<td>Run production executions</td>
</tr>
<tr>
<td>Small &amp; medium compilations</td>
<td>Copy large amount of files</td>
</tr>
<tr>
<td>Submit jobs to batch system</td>
<td>Long and heavy load graphical interfaces</td>
</tr>
<tr>
<td>Check results and prepare scripts</td>
<td></td>
</tr>
</tbody>
</table>
Compilers

- Intel and GNU compiler suites available
- Intel compilers available in:
  - login1
  - Interactive nodes ($salloc -p interactive)
- Several versions, managed by modules
  - Fortran, C, C++
  - Intel (licensed)
  - GCC (Free Software)
- MPI compilation also managed by modules through wrappers
  - mpicc, mpifort...
Compiler optimization drawbacks

• Each software is different, but:
  – Intel: up to 20% performance increase in some applications
    • Linking mkl libraries normally boosts performance
  – Static compilation sometimes runs faster

• Optimization drawbacks
  – Over optimization may result in numeric error
Module Environment (I)

- Open Source project
- System used by Marenosstrum to manage all installed software
- Environment variables and software dependencies management
- Several versions of same program side-to-side (/gpfs/apps only)
- If you need additional software, you can ask support to install it as a module
- Typical dependencies:
  - MPI libraries
  - Mathematical libraries
## Module Environment (II)

### Module commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Option</th>
<th>Example</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>avail</td>
<td>[program]</td>
<td>module avail</td>
<td>List modules available</td>
</tr>
<tr>
<td>list</td>
<td></td>
<td>module list</td>
<td>List loaded modules</td>
</tr>
<tr>
<td>purge</td>
<td></td>
<td>module purge</td>
<td>Unload all modules</td>
</tr>
<tr>
<td>load</td>
<td>&lt;program[/version]&gt;</td>
<td>module load gcc/5.1.0</td>
<td>Load a module</td>
</tr>
<tr>
<td>switch</td>
<td>&lt;old&gt; &lt;new&gt;</td>
<td>module switch intel gcc</td>
<td>Change a module by another</td>
</tr>
</tbody>
</table>
Batch System

- MareNostrum IV uses Platform SLURM as batch system

- Benefits of using jobscripts
  - Defines resources needed
  - Reusable
  - Documents needs and requests
  - Jobscripts are shellscripts with special markings

- Each submission is a job

```bash
bsc99526@login3:/CLUSTER_TESTS/MN4> sbatch slurm-test.sh
Submitted batch job 1405572
bsc99526@login3:/CLUSTER_TESTS/MN4> squeue -u $USER

  JOBID  PARTITION  NAME   USER   ST  TIME  NODES  Nodelist(REASON)   
  1405572  main      test bsc99526  PD  0:00  1 (Resources)
```
SLURM commands

• Submit a job defined in job_script.cmd
  - sbatch job_script.cmd

```bash
bsc99526@login3:~/CLUSTER_TESTS/MN4> sbatch slurm-test.sh
Submitted batch job 1405572
```

• Check status of jobs submitted:
  - User's: squeue

```bash
bsc99526@login3:~/CLUSTER_TESTS/MN4> squeue -u $USER

JOBID PARTITION   NAME      USER ST   TIME NODES NODELIST(REASON)
1405572  main      test bsc99526 PD   0:00  1 (Resources)
```

• Cancel a job:
  - scancel JobID

```bash
bsc99526@login3:~/CLUSTER_TESTS/MN4> scancel 1405572
```
# SLURM Common Parameters

<table>
<thead>
<tr>
<th>Option</th>
<th>Comment</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>-n</td>
<td>--ntasks</td>
<td>Number of tasks</td>
</tr>
<tr>
<td>-t</td>
<td>--time</td>
<td>Wallclock limit</td>
</tr>
<tr>
<td>-J</td>
<td>--job-name</td>
<td>Job name</td>
</tr>
<tr>
<td>-o</td>
<td>--output</td>
<td>Output file</td>
</tr>
<tr>
<td>-e</td>
<td>--error</td>
<td>Error file</td>
</tr>
<tr>
<td>--qos</td>
<td>Queue</td>
<td>#SBATCH --qos debug</td>
</tr>
<tr>
<td>--exclusive</td>
<td>Exclusive mode</td>
<td>#SBATCH --exclusive</td>
</tr>
<tr>
<td>-D</td>
<td>--workdir</td>
<td>Current working dir</td>
</tr>
<tr>
<td>--reservation</td>
<td>Reservation</td>
<td>#SBATCH --reservation reserv_name</td>
</tr>
</tbody>
</table>
**SLURM Extra Parameters: Process layout**

- How to define specific load balance configurations:

<table>
<thead>
<tr>
<th>Option</th>
<th>Comment</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>--ntasks-per-core</td>
<td>Tasks per core</td>
<td>#SBATCH --ntasks-per-core 1</td>
</tr>
<tr>
<td>--ntasks-per-node</td>
<td>Tasks per node</td>
<td>#SBATCH --ntasks-per-node 48</td>
</tr>
<tr>
<td>-c</td>
<td>--cpus-per-task</td>
<td>Cpus per task</td>
</tr>
</tbody>
</table>

**Generic MNIV example:**

```
#SBATCH --ntasks-per-core 1
#SBATCH --ntasks-per-node 48
```
SLURM Extra Parameters: Memory layout

2 nodetypes...

• **HIGH MEMORY NODES**
  - Total of 384GBytes per node (8G per core)
  - Only 216 nodes available
  - `#SBATCH –constraint=highmem`
  - Requesting them will make you wait more in the queue

• **LOW MEMORY NODES**
  - Total of 96GBytes per node (2G per core)
  - Default nodes
Job queues

- Jobs are assigned to queues (QoS)
  - Default queue automatically selected.
- Specify when special need: debug, interactive, graphical...
  - Different queues have different limits and goals
- Check your available queues and their limits:
  - bsc_queues

```bash
glogin3:/gpfs/home/bsc99/bsc99526/CLUSTER_TESTS/MN4> bsc_queues
```

<table>
<thead>
<tr>
<th>QUEUE NAME</th>
<th>MAX TIME</th>
<th>MAX PROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>debug</td>
<td>02:00:00</td>
<td>768</td>
</tr>
<tr>
<td>interactive</td>
<td>02:00:00</td>
<td>4</td>
</tr>
<tr>
<td>prace</td>
<td>3-00:00:00</td>
<td>19200</td>
</tr>
</tbody>
</table>

- Example: #SBATCH --qos debug
Job Examples: Sequential

- Sequential example...

```bash
#!/bin/bash
#SBATCH -n 1
#SBATCH -o %J.out
#SBATCH -e %J.err
#SBATCH -t 01:00

hostname
```
Job Examples: Threaded

- Threaded example with (OpenMP, pthreads, ...)

```bash
#!/bin/bash

#SBATCH --ntasks=1  # Run a single task
#SBATCH --cpus-per-task=6  # Number of CPU cores per task
#SBATCH -o %j.out  # File to which STDOUT will be written, %j inserts jobid
#SBATCH -e %j.err  # File to which STDERR will be written, %j inserts jobid
#SBATCH -t 01:00  # Runtime in D-HH:MM, minimum of 01:00 minutes
...
#SBATCH --exclusive  # Allocate full node (*)
...
export OMP_NUM_THREADS=6
...
```
Job Examples: Typical MPI

• MPI example with (multiple nodes, OpenMPI)

```bash
#!/bin/bash
#SBATCH -n 96
#SBATCH -o %j.out
#SBATCH -e %j.err
#SBATCH -t 01:00
module purge
module load openmpi
mpirun ...
```
Job Examples: MPI + OpenMP

- MPI + Threads example...

```bash
#!/bin/bash
#SBATCH -n 96
#SBATCH -o %j.out
#SBATCH -e %j.err
#SBATCH --cpus-per-task=4
#SBATCH -t 01:00
export OMP_NUM_THREADS=4
module purge
module load openmpi
mpirun ...
```
Support Contact practices

• When contacting support remember this:
  – Specify
    • Job Ids
    • Software version
    • Environment (if applies)
    • Machine
    • Username
    • Exact steps that lead to the issue
    • Error messages

  – **Do not** take for granted that we know what you know, want or need. Effective communication results in a **faster resolving time**.

• We don't know who you are but we will try our best to help you
• We have no favorites
Hands-on 2: MareNostrum Usage
Thank you

support@bsc.es