Hands-on 1:

Getting Started with Marenostrum III

SUPERCOMPUTERS ARCHITECTURE

MASTER IN INNOVATION AND RESEARCH IN INFORMATICS

Barcelona, Fall 2015

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Hands-on 1

Getting Started with Marenostrum III

SUPERCOMPUTERS ARCHITECTURE
Master in innovation and research in informatics
(Specialization High Performance Computing)
UPC Barcelona Tech & Barcelona Supercomputing Center

Version 2.0 - 29 September 2015

Professor: Jordi Torres jordi.torres@bsc.es www.JordiTorres.eu

Table of Contents

1 Hands-on description

This hands-on is intended to help the student to get started with a Supercomputer. In our course we will use the Marenostrum III supercomputer from Barcelona Supercomputing Center – Centro Nacional de Supercomputation located in UPC campus.

This Hands-On starts with the steps required to obtain an account with the supercomputer. Later we will present the *Data Transfer* machines available for BSC users and the special data transfer commands required to run batch jobs. It also shows the structure of the GPFS shared file systems and other File Systems at BSC facilities. Finally the student will write and run their first job in Marenostrum.

This hands-on will be guided by the teacher during the lab session.

2 Marenostrum III System Overview

MareNostrum III (MN III) is a supercomputer based on Intel SandyBridge processors, iDataPlex Compute Racks, a Linux Operating System and an Infiniband interconnection.

The current Peak Performance is 1.1 Petaflops. The total number of processors is 48,896 Intel SandyBridge-EP E5–2670 cores at 2.6 GHz (3,056 compute nodes) with 95.5 TB of main memory. See below a summary of the system:

- 36 iDataPlex compute racks. Each one composed of:
  - 84 IBM dx360 M4 compute nodes
  - 4 Mellanox 36-port Managed FDR10 IB Switches
  - 2 BNT RackSwitch G8052F (Management Network)
  - 2 BNT RackSwitch G8052F (GPFS Network)
  - 4 Power Distribution Units
- Each IBM dx360 M4 node contains:
  - 2x E5–2670 SandyBridge-EP 2.6GHz cache 20MB 8-core
  - 500GB 7200 rpm SATA II local HDD
  - 8x 4G DDR3–1600 DIMMs (2GB/core) Total: 32GB/node
  - Dual-port Infiniband QDR/FDR10 Mezzazine Card

- 1.9 PB of GPFS disk storage
- Interconnection Networks
  - Infiniband Mellanox FDR10: High bandwidth network used by parallel applications com- munications (MPI)
  - Gigabit Ethernet: 10GbitEthernet network used by the GPFS Filesystem.
- Operating System: Linux - SuSe Distribution 11 SP3

3 How to acces Marenostrum III

3.1 How to get your account

Each student/group (to be decided according the number of students) will have an account "sam140**". This will be assigned by the teacher at the beginning of the first session lab.

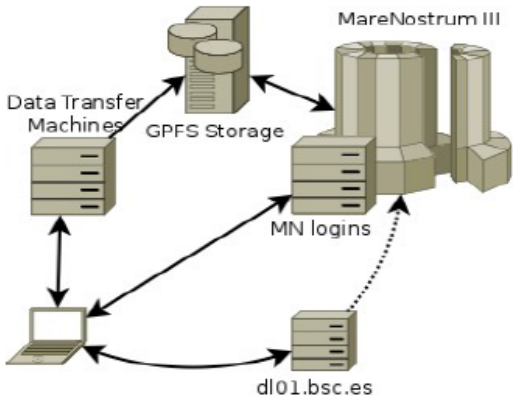
Exercise 1: Send an email to support@bsc.es (with CC *jordi.torres@bsc.es*) including

(a) As a subject "Master course SA-MIRI: account information sam140**" ("**" refers the number assigned to your group),
(b) Name of all group members
(c) Contact Email of all group members

The password will be sent in a separate mail with no subject unless you already had one assigned by the teacher.

3.2 Change your password

For security reasons the password must be changed the first time every user accesses the machine. BSC uses a centralized passwords management system in order to access all BSC resources. In order to change your password, you have to login **dl01.bsc.es** server from your local machine with the same username and password as in the cluster.



Then, you have to use the 'passwd' command.

```
% ssh username@dl01.bsc.es

username@dlogin1:~>
passwd Changing
password for username.
Old Password:

New Password:
Reenter New
Password: Password
changed.
```

Exercise 2: Change your password.

The new password will become effective 10 minutes after the change.

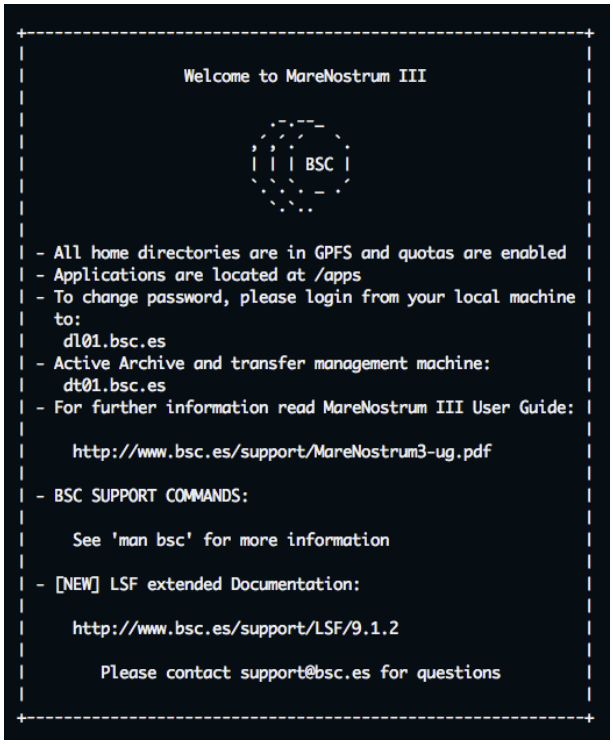
3.3 Connecting to MareNostrum III

Once you have a login username and its associated password you can get into the cluster through one of the following login nodes in Marenostrum III:

- mn1.bsc.es
- mn2.bsc.es
- mn3.bsc.es

You must use Secure Shell (ssh) tools to login into or transfer files into the cluster in order to enable secure logins over an insecure network. You will find a ssh client in the lab terminal (warning: BSC do not accept incoming connections from protocols like telnet, ftp, rlogin, rcp, or rsh commands). Once you have logged into the cluster you cannot make outgoing connections for security reasons.

Once connected to the machine, you will be presented with a UNIX shell prompt and you will normally be in your home ($HOME) directory (In this lab we assume that you are not new to UNIX environments).



When accessing to Marenostrum for the first time, an error message regarding ssh keys may appear. If this is the case you have to erase (mn1,mn2,mn3) entries from $HOME/.ssh/known_hosts!

Exercise 3: List the files in your home directory.

Congratulations, you are inside Marenostrum. Welcome on board!

4 Marenostrum File Systems

Each user has several areas of disk space for storing files. These areas may have size or time limits. There are 3 different types of storage available inside a node:

- *Root filesystem*: Is the filesystem where the operating system resides
- *GPFS filesystems*: GPFS is a distributed networked filesystem which can be accessed from all the nodes and Data Transfer Machine (discussed previously)
- *Local hard drive*: Every node has an internal hard drive

4.4 Root Filesystem

The root file system, where the operating system is stored doesn't reside in the node, this is a NFS filesystem mounted from one of the servers.

As this is a remote filesystem only data from the operating system should to reside in this filesystem. The use of /tmp for temporary user data is NOT permitted. The local hard drive can be used for this purpose as you could read in Local Hard Drive (presented later).

4.5 GPFS Filesystem

The IBM General Parallel File System (GPFS) is a high-performance shared-disk file system providing fast, reliable data access from all nodes of the cluster to a global filesystem. GPFS allows parallel applications simultaneous access to a set of files (even a single file) from any node that has the GPFS file system mounted while providing a high level of control over all file system operations. In addition, GPFS can read or write large blocks of data in a single I/O operation, thereby minimizing overhead.

The following are the GPFS filesystems available in the machine from all nodes:

- */apps*: Over this filesystem will reside the applications and libraries that have already been installed on the machine. Take a look at the directories to know the applications available for general use.

- */gpfs/home*: This filesystem has the home directories of all the users, and when you log in you start in your home directory by default. Every user will have their own home directory to store own developed sources and their personal data. A default quota will be enforced on all users to limit the amount of data stored there. Also, it is highly discouraged to run jobs from this filesystem. Please run your jobs on your group's /gpfs/projects or /gpfs/scratch instead.
- */gpfs/projects*: In addition to the home directory, there is a directory in */gpfs/projects* for each group of users. For instance, the group bsc01 will have a */gpfs/projects/bsc01* directory ready to use. This space is intended to store data that needs to be shared between the users of the same group or project. A quota (section 7) per group will be enforced depending on the space assigned by Access Committee at Marenostrum. It is the project's manager responsibility to determine and coordinate the better use of this space, and how it is distributed or shared between their users.
- */gpfs/scratch*: Each user will have a directory over */gpfs/scratch*. Its intended use is to store temporary files of your jobs during their execution. A quota per group will be enforced depending on the space assigned.

Exercise 4: Using linux commands (e.g. cd, ls, …) determine how many users and apps there are at MN III.

4.6 Local Hard Drive

Every node has a local hard drive that can be used as a local scratch space to store temporary files during executions of one of your jobs. This space is mounted over /scratch/tmp directory and pointed out by $TMPDIR environment variable. The amount of space within the /scratch filesystem is about 500 GB. All data stored in these local hard drives at the compute nodes will not be available from the login nodes.
Local hard drive data is not automatically removed, so each job has to remove its data before finishing each job.

Exercise 5: Inspect the content of your local hard drive. Has the previous user removed his/her data?

5 Transferring files in Marenostrum III

There are two ways to copy files from/to the Cluster:

- Direct scp or sftp to the login nodes
- Using a Data transfer Machine which shares all the GPFS filesystem for transferring large files

5.7 Direct copy to the login nodes

As stated previously, no connections are allowed from inside the cluster to the outside world, so all scp and sftp commands have to be executed from your local machines and never from the cluster.

Here there are some examples of each of these tools transferring small files to the cluster:

```
localsystem$ scp localfile
username@mn1.bsc.es: username's
password:

localsystem$  sftp
username@mn1.bsc.es username's
password:

sftp> put localfile
```

These are the ways to retrieve files from the login nodes to your local machine:

```
localsystem$ scp username@mn1.bsc.es:remotefile
localdir username's password:

localsystem$  sftp
username@mn1.bsc.es username's
password:

sftp> get remotefile
```

Exercise 6: Use one of these tools for transferring a file between your local machine and Marenostrum.

5.8 Data Transfer Machine

Marenostrum environment provides special machines for file transfer (required for large amounts of data). These machines are dedicated to Data Transfers and are accessible through ssh with the same username and password as MareNostrum account. They are:

- dt01.bsc.es
- dt02.bsc.es

These machines share the GPFS filesystem with all other BSC HPC machines. Besides scp and sftp, they allow some other useful transfer protocols:

- BBCP (out of the scope of this master course)

```
bbcp -V -z <USER>@dt01.bsc.es:<FILE> <DEST>
bbcp -V <ORIG>  <USER>@dt01.bsc.es:<DEST>
```

- FTPS (out of the scope of this master course)

```
gftp-text
ftps://<USER>@dt01.bsc.es get
<FILE>

put <FILE>
```

- GRIDFTP (only accessible from dt02.bsc.es) (out of the scope of this hands-on)

6 Active Archive Management

Active Archive (AA) is a mid-long term storage filesystem that provides more than 3 PB of total space. You can access AA from the Data Transfer Machine (dt01.bsc.es and dt02.bsc.es) under /gpfs/archive/your_group.

NOTE: There is no backup of this filesystem. The user is responsible for adequately managing the data stored in it.

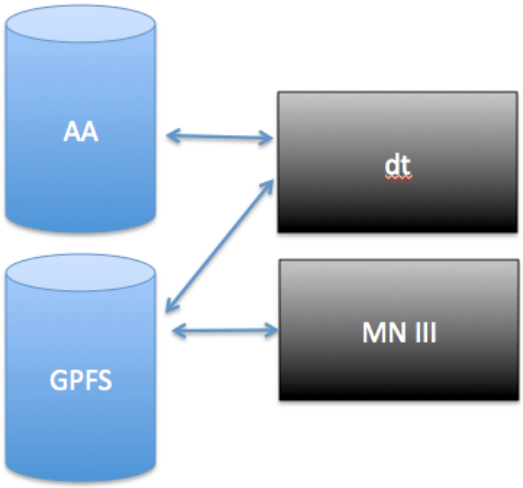


Figure 1: General scheme of storage systems access

Exercise 7: Inspect the "/gpfs/archive" directory. Describe what happened.

Important: The problem is that AA is only mounted in Data Transfer Machine. Therefore if you wish to navigate through AA directory tree you have to login into dt01.bsc.es.

To move or copy from/to Active Archive you have to use Marenostrum special commands:

- dtcp, dtmv, dtrsync, dttar

These commands submit a job into a special class performing the selected command. Their syntax is the same as the shell command without 'dt' prefix (cp, mv, rsync, tar).

- dtq, dtcancel

  dtq shows all the transfer jobs that belong to you. (works like mnq)

dtcancel works like mncancel (see below) for transfer jobs.

- dttar: submits a tar command to queues.

Example: Taring data from /gpfs/to /gpfs/archive

```
% dttar -cvf  /gpfs/archive/usertest/outputs.tar ~/OUTPUTS
```

- *dtcp:* submits a cp command to queues. Remember to delete the data in the source filesystem once copied to AA to avoid duplicated data.

```
#Example: Copying data from /gpfs to /gpfs/archive
% dtcp -r  ~/OUTPUTS /gpfs/archive/usertest/
```

```
#Example: Copying data from /gpfs/archive to /gpfs
% dtcp -r /gpfs/archive/usertest/OUTPUTS ~/
```

- *dtmv:* submits a mv command to queues.

```
#Example: Moving data from /gpfs to /gpfs/archive
% dtmv ~/OUTPUTS /gpfs/archive/usertest/
```

```
#Example: Moving data from /gpfs/archive to /gpfs
% dtmv /gpfs/archive/usertest/OUTPUTS ~/
```

It is important to note that these kind of jobs can be submitted from both the 'login' nodes (automatic file management within a production job) and 'dt01.bsc.es' machine. Remember that AA is only mounted in Data Transfer Machine. Therefore if you wish to navigate through AA directory tree you have to login into dt01.bsc.es

7 Quotas

The quotas are the amount of storage available for a user or a groups' users. You can picture it as a small disk readily available to you. A default value is applied to all users and groups and cannot be outgrown.

You can inspect your quota anytime you want using the following commands from inside each filesystem:

```
% quota
% quota -g <GROUP>
% bsc_quota
```

The first command provides the quota for your user and the second one provides the quota for your group, showing the totals of both granted and used quota. The third command provides a more readable output for the quota.

Exercise 8: Check your assigned quota.

8 Basic C Compilers

8.1 Getting started

In the Marenostrum cluster you can find these C/C++ compilers : icc /icpc -> Intel C/C++ Compilers

```
% man icc
% man icpc
```

gcc /g++ -> GNU Compilers for C/C++

```
% man gcc
% man g++
```

All invocations of the C or C++ compilers follow these suffix conventions for input files:

.C, .cc, .cpp, or .cxx -> C++ source file.
.c -> C source file
.i -> preprocessed C source file
.so -> shared object file
.o -> object file for ld command
.s -> assembler source file

By default, the preprocessor is run on both C and C++ source files.

Exercise 9: Create a “Hello World” program, compile it and run it.

Use the editor vi and create the program. If you are not familiar with vi editor you can use the NEDIT editor also available in Marenostrum using this command:

```
% /apps/NEDIT/5.5/bin/nedit &
```

Note: to launch NEDIT in your local screen use the flag –X in the ssh command.

8.2 Serial matrix-vector multiplication example

Exercise 10 – OPTIONAL: Implement a serial matrix-vector multiplication C program (mat_vect_mult.c) using one-dimensional arrays to store the vectors and the matrix. As an input the mat_vect_mult program will receive the dimensions of the matrix (m = number of rows, n= number of columns). The values of the matrix can be obtained directly by the rand() function. The output will be the product vector `y = Ax.`

This code will be used in a next hands-on.

9 Additional information

All BSC commands (quota, dtls, …) has man page (man bsc_commands (man dtcommands). Important additional information can be found at FAQ section of www.bsc.es/user-support/

10 Lab Report

You have one week to deliver a report with the answers to the exercises.

Acknowledgement: Part of this hands-on is based on "Marenostrum III User's Guide". Special thanks to David Vicente and Miguel Bernabeu from BSC Operations department for his invaluable help preparing this hands-on.