Knights Landing CTE User's Guide

Barcelona Supercomputing Center

May 4, 2021

Contents

1 Introduction

This user's guide for the CTE Intel Xeon Phi Knights Landing cluster 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 many of the standard features of supercomputing as the Unix operating system.

Here you can find most of the information you need to use our computing resources and the technical documentation about the machine. Please read carefully this document and if any doubt arises do not hesitate to contact us (Getting help (chapter 11)).

2 System Overview

CTE-KNL is a cluster based on Intel Xeon Phi Knights Landing processors, a Linux Operating System and an Intel OPA interconnection.

It has the following configuration:

- Login node ksmp (previously smp1 from MareNostrum 3)
  - 80 cores Intel(R) Xeon(R) CPU E7- 8850 @ 2.00GHz (8 NUMA nodes)
  - 2 TB of main memory
  - 900 GB as local storage (RAID 1)
  - GPFS via two fiber links 10 GBit
- 16 compute nodes
  - 1 Intel(R) Xeon Phi(TM) CPU 7230 @ 1.30GHz 64-core processor
  - 96 GB main memory distributed in 6x 16GB DDR4 @ 1200 MHz dimms (90 GB/s)
  - 16 GB high bandwidth memory distributed in 8x 2GB MCDRAM @ 7200 Mhz dimms (480 GB/s)
  - 120 GB SSD as local storage
  - Peak Performance 1.8 TFlops
  - 100 Gbits/s Omni-Path interface
  - GPFS via ethernet 1 GBit

Hyperthreading is currently disabled on these machines, therefore 64 cores is the maximum per node

Currently all nodes are configured in Quadrant Cluster Mode, for Memory Mode please check (Job directives (section 9.3)). For the time being it is not possible to change this configuration on the fly. If some nodes are required to be in other mode please send a request to support@bsc.es and this will be treated on a case to case basis.

The operating system is SUSE Linux Enterprise Server 12 SP2 for both configurations.

3 Compiling applications

Please note that for optimal performance you will need to cross compile for the **AVX–512 instructions** available in the KNLs.

For compiling applications the system provides GCC version 4.8.5 and Intel Parallel Studio XE 2017.1 is available in /apps and via modules.

More information can be found in the PRACE KNL Best Practice Guide[1]

4 Interconnect Intel Omni-Path

The cluster is equipped with a new generation of interconnect fabric, the Intel Omni-Path Architecture (Intel OPA[2]).

[1] http://www.prace-ri.eu/best-practice-guide-knights-landing-january-2017/

[2] http://www.intel.com/content/www/us/en/high-performance-computing-fabrics/omni-path-architecture-fabric-overview.html

Compiler	Suggested Flags
Intel C compiler	-O3 -xMIC-AVX512 -fma -align -finline-functions
Intel C++ compiler	-std=c11 -O3 -xMIC-AVX512 -fma -align -finline-functions
Intel Fortran compiler	-O3 -xMIC-AVX512 -fma -align array64byte -finline-functions
GCC compiler	-march=knl -O3 -mavx512f -mavx512pf -mavx512er -mavx512cd -mfma -malign-data=cacheline -finline-functions
G++ compiler	-std=c11 -march=knl -O3 -mavx512f -mavx512pf -mavx512er -mavx512cd -mfma -malign-data=cacheline -finline-function
Gfortran compiler	-O3 -march=knl -mavx512f -mavx512pf -mavx512er -mavx512cd -mfma -malign-data=cacheline -finline-functions

Each KNL node has a PCI express interface and they are all connected to a single OPA switch. The interface in the nodes is named ib0 and identified as InfiniBand as by the Linux kernel, although they really are OPA interfaces.

By default with Intel MPI jobs are using the Omni-Path network. You can also switch between the OPA and Ethernet interfaces via MPI environment settings.

- Intel MPI - Ethernet `export I_MPI_FABRICS_LIST=tcp`
- Intel MPI - Omni-Path `export I_MPI_FABRICS_LIST=tmi`

You can find more information on fabric selection here[3]

```
[knl05 ~]$ ibstat
CA 'hfi1_0'
    CA type:
    Number of ports: 1
    Firmware version:
    Hardware version: 11
    Node GUID: 0x0011750101778494
    System image GUID: 0x0011750101778494
    Port 1:
        State: Active
        Physical state: LinkUp
        Rate: 100
        Base lid: 14
        LMC: 0
        SM lid: 1
        Capability mask: 0x00410020
        Port GUID: 0x0011750101778494
        Link layer: InfiniBand
```

5 High Bandwidth Memory MCDRAM

The KNL processors have an additional memory of 16 GB that can be used to accelerate applications if used. It can be configured in different ways. Currently all nodes are configured in **cache mode** and the operating system automatically uses this memory to cache frequently used data.

Therefore there is only one NUMA node visible with all the CPU cores and DDR4 memory:

```
numactl -H
available: 1 nodes (0)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
node 0 size: 96523 MB
node 0 free: 92647 MB
node distances:
node   0
  0:  10
```

[3] https://software.intel.com/en-us/node/535584

For the time being it is not possible to change this configuration on the fly. If some nodes are required to be in flat mode please send a request to support@bsc.es and this will be treated on a case to case basis.

6 Connecting to CTE-KNL

The first thing you should know is your username and password. Once you have a login and its associated password you can get into the cluster through the following login node:

- klogin1.bsc.es

This will provide you with a login shell in the SMP node. There you can compile and prepare your applications.

You must use Secure Shell (ssh) tools to login into or transfer files into the cluster. We 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.

6.1 Password Management

In order to change the password, you have to login to a different machine (dt01.bsc.es). This connection must be established from your local machine.

```
% ssh -l username dt01.bsc.es

username@dtransfer1:~> passwd
Changing password for username.
Old Password:
New Password:
Reenter New Password:
Password changed.
```

Mind that the password change takes about 10 minutes to be effective.

7 File Systems

IMPORTANT: It is your responsibility as a user of our facilities to backup all your critical data. *We only guarantee a daily backup of user data under /gpfs/home. Any other backup should only be done exceptionally under demand of the interested user.*

Each user has several areas of disk space for storing files. These areas may have size or time limits, please read carefully all this section to know about the policy of usage of each of these filesystems. There are 3 different types of storage available inside a node:

- *GPFS filesystems*: GPFS is a distributed networked filesystem which can be accessed from all the nodes and Data Transfer Machine (section 8.1)
- *Local hard drive*: Every node has an internal hard drive
- *Root filesystem*: Is the filesystem where the operating system resides

7.1 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.

An incremental backup will be performed daily only for /gpfs/home.

These 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 (section 7.5) 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 7.5) per group will be enforced depending on the space assigned by Access Committee. 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 (section 7.5) per group will be enforced depending on the space assigned.

7.2 Active Archive - HSM (Tape Layer)

Active Archive (AA) is a mid-long term storage filesystem that provides 15 PB of total space. You can access AA from the Data Transfer Machine (section 8.1) (dt01.bsc.es and dt02.bsc.es) under /gpfs/archive/hpc/your_group.

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

Hierarchical Storage Management (HSM) is a data storage technique that automatically moves data between high-cost and low-cost storage media. At BSC, the filesystem using HSM is the one mounted at /gpfs/archive/hpc, and the two types of storage are GPFS (high-cost, low latency) and Tapes (low-cost, high latency).

HSM System Overview

Hardware

- IBM TS4500 with 10 Frames
- 6000 Tapes 12TB LTO8
- 64 Drives
- 8 LC9 Power9 Servers

Software

- IBM Spectrum Archive 1.3.1
- Spectrum Protect Policies

Functioning policy and expected behaviour

In general, this automatic process is transparent for the user and you can only notice it when you need to access or modify a file that has been migrated. If the file has been migrated, any access to it will be delayed until its content is retrieved from tape.

- Which files are migrated to tapes and which are not?

Only the files with a size between 1GB and 12TB will be moved (migrated) to tapes from the GPFS disk when no data access and modification have been done for a period of 30 days.

- Working directory (under which copies are made)

```
/gpfs/archive/hpc
```

- What happens if I try to modify/delete an already migrated file?

From the user point of view, the deletion will be transparent and have the same behaviour. On the other hand, it is not possible to modify a migrated file; in that case, you will have to wait for the system to retrieve the file and put it back on disk.

- What happens if I'm close to my quota limit?

If there is not enough space to recover a given file from tape, the retrieve will fail and everything will remain in the same state as before, that is, you will continue to see the file on tape (in the "migrated" state).

- How can I check the status of a file?

You can use the *hsmFileState* script to check if the file is *resident* on disk or has been *migrated* to tape..

Examples of use cases

```
$ hsmFileState file_1MB.dat
resident -rw-rw-r-- 1 user group 1048576 mar 12 13:45 file_1MB.dat

$ hsmFileState file_10GB.dat
migrated -rw-rw-r-- 1 user group 10737418240 feb 12 11:37 file_10GB.dat
```

7.3 Local Hard Drive

Every node has a local solid-state 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 /tmp directory. The amount of space within the /tmp filesystem is about 80 GB. All data stored in these local solid-state drive at the compute nodes will not be available from the login nodes. **Local solid-state drive data are not automatically removed, so each job has to remove its data before finishing**.

7.4 Root Filesystem

The root file system, where the operating system is stored has its own partition.

There is a separate partition of the local hard drive mounted on /tmp that can be used for storing user data as you can read in Local Hard Drive (section 7.3).

7.5 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 command from inside each filesystem:

```
% bsc_quota
```

The command provides a readable output for the quota.

If you need more disk space in this filesystem or in any other of the GPFS filesystems, the responsible for your project has to make a request for the extra space needed, specifying the requested space and the reasons why it is needed. For more information or requests you can Contact Us (chapter 11).

8 Data management

8.1 Transferring files

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

Direct copy to the login nodes.

As said before 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. The usage examples are in the next section.

On a Windows system, most of the secure shell clients come with a tool to make secure copies or secure ftp's. There are several tools that accomplish the requirements, please refer to the Appendices (chapter 13), where you will find the most common ones and examples of use.

Data Transfer Machine

We provide special machines for file transfer (required for large amounts of data). These machines are dedicated to Data Transfer and are accessible through ssh with the same account credentials as the cluster. 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:

- scp

```
localsystem$ scp localfile username@dt01.bsc.es:
username's password:

localsystem$ scp username@dt01.bsc.es:remotefile localdir
username's password:
```

- rsync

```
localsystem$ rsync -avzP localfile_or_localdir username@dt01.bsc.es:
username's password:

localsystem$ rsync -avzP username@dt01.bsc.es:remotefile_or_remotedir localdir
username's password:
```

- sftp

```
localsystem$ sftp username@dt01.bsc.es
username's password:
sftp> get remotefile

localsystem$ sftp username@dt01.bsc.es
username's password:
sftp> put localfile
```

- BBCP

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

- GRIDFTP (only accessible from dt02.bsc.es)
- SSHFTP

```
globus-url-copy -help
globus-url-copy -tcp-bs 16M -bs 16M -v -vb your_file sshftp://your_user@dt01.bsc.es/~/
```

Data Transfer on the PRACE Network

PRACE users can use the 10Gbps PRACE Network for moving large data among PRACE sites. To get access to this service it's required to contact "support@bsc.es" requesting its use, providing the local IP of the machine from where it will be used.

The selected data transfer tool is Globus/GridFTP[4] which is available on dt02.bsc.es
In order to use it, a PRACE user must get access to dt02.bsc.es:

```
% ssh -l pr1eXXXX dt02.bsc.es
```

Load the PRACE environment with 'module' tool:

```
% module load prace globus
```

Create a proxy certificate using 'grid-proxy-init':

```
% grid-proxy-init
Your identity: /DC=es/DC=irisgrid/O=bsc-cns/CN=john.foo
Enter GRID pass phrase for this identity:
Creating proxy ..... Done
Your proxy is valid until: Wed Aug  7 00:37:26 2013
pr1eXXXX@dtransfer2:~>
```

The command 'globus-url-copy' is now available for transferring large data.

```
globus-url-copy [-p <parallelism>] [-tcp-bs <size>] <sourceURL> <destURL>
```

Where:

- -p: specify the number of parallel data connections should be used (recommended value: 4)
- -tcp-bs: specify the size (in bytes) of the buffer to be used by the underlying ftp data channels (recommended value: 4MB)
- Common formats for sourceURL and destURL are:
  - file:///(on a local machine only) (e.g. file:///home/pr1eXX00/pr1eXXXX/myfile)
  - gsiftp://(e.g. gsiftp://supermuc.lrz.de/home/pr1dXXXX/mydir/)
  - remember that any url specifying a directory must end with /.

All the available PRACE GridFTP endpoints can be retrieved with the 'prace_service' script:

```
% prace_service -i -f bsc
gftp.prace.bsc.es:2811
```

More information is available at the PRACE website[5]

[4] http://www.globus.org/toolkit/docs/latest-stable/gridftp/
[5] http://www.prace-ri.eu/Data-Transfer-with-GridFTP-Details

8.2 Active Archive Management

To move or copy from/to AA you have to use our special commands, available in dt01.bsc.es and dt02.bsc.es or any other machine by loading "transfer" module:

- dtcp, dtmv, dtrsync, dttar

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

- dtq, dtcancel

```
dtq
```

dtq shows all the transfer jobs that belong to you, it works like squeue in SLURM.

```
dtcancel <job_id>
```

dtcancel cancels the transfer job with the job id given as parameter, it works like scancel in SLURM.

- *dttar:* submits a tar command to queues. Example: Taring data from /gpfs/to /gpfs/archive/hpc

```
% dttar -cvf /gpfs/archive/hpc/group01/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/hpc
% dtcp -r ~/OUTPUTS /gpfs/archive/hpc/group01/
```

```
# Example: Copying data from /gpfs/archive/hpc to /gpfs
% dtcp -r /gpfs/archive/hpc/group01/OUTPUTS ~/
```

- *dtrsync:* submits a rsync 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/hpc
% dtrsync -avP ~/OUTPUTS /gpfs/archive/hpc/group01/
```

```
# Example: Copying data from /gpfs/archive/hpc to /gpfs
% dtrsync -avP /gpfs/archive/hpc/group01/OUTPUTS ~/
```

- *dtsgrsync:* submits a rsync command to queues switching to the specified group as the first parameter. If you are not added to the requested group, the command will fail. Remember to delete the data in the source filesystem once copied to the other group to avoid duplicated data.

```
# Example: Copying data from group01 to group02
% dtsgrsync group02 /gpfs/projects/group01/OUTPUTS /gpfs/projects/group02/
```

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

```
# Example: Moving data from /gpfs to /gpfs/archive/hpc
% dtmv ~/OUTPUTS /gpfs/archive/hpc/group01/
```

```
# Example: Moving data from /gpfs/archive/hpc to /gpfs
% dtmv /gpfs/archive/hpc/group01/OUTPUTS ~/
```

Additionally, these commands accept the following options:

```
--blocking: Block any process from reading file at final destination until transfer completed.

--time: Set up new maximum transfer time (Default is 18h).
```

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. AA is only mounted in Data Transfer Machine (section 8.1). Therefore if you wish to navigate through AA directory tree you have to login into dt01.bsc.es

8.3 Repository management (GIT/SVN)

There's no outgoing internet connection from the cluster, which prevents the use of external repositories directly from our machines. To circumvent that, you can use the "sshfs" command in your local machine.

Doing that, you can mount a desired directory from our GPFS filesystem in your local machine. That way, you can operate your GPFS files as if they were stored in your local computer. That includes the use of git, so you can clone, push or pull any desired repositories inside that mount point and the changes will transfer over to GPFS.

Setting up sshfs

- Create a directory inside your local machine that will be used as a mount point.
- Run the following command below, where the local directory is the directory you created earlier. Note that this command mounts your GPFS home directory by default.

```
sshfs -o workaround=rename <yourHPCUser>@dt01.bsc.es: <localDirectory>
```

- From now on, you can access that directory. If you access it, you should see your home directory of the GPFS filesystem. Any modifications that you do inside that directory will be replicated to the GPFS filesystem inside the HPC machines.
- Inside that directory, you can call "git clone", "git pull" or "git push" as you please.

9 Running Jobs

Slurm is the utility used for batch processing support, so all jobs must be run through it. This section provides information for getting started with job execution at the Cluster.

9.1 Submitting jobs

The method for submitting jobs is to use the SLURM *sbatch* directives directly.

A job is the execution unit for SLURM. A job is defined by a text file containing a set of directives describing the job's requirements, and the commands to execute.

In order to ensure the proper scheduling of jobs, there are execution limitations in the number of nodes and cpus that cna be used at the same time by a group. You may check those limits using command 'bsc_queues'. If you need to run an execution bigger than the limits already granted, you may contact support@bsc.es.

Available partitions

The CTE-KNL cluster is comprised of both the Knights Landing Compute Nodes and the SMP login node. You can submit jobs to either the KNL partition (15 nodes) or the SMP partition (up to 74 cores).

```
# sbatch
#SBATCH --partition=knl
OR
#SBATCH --partition=smp
```

SBATCH commands

These are the basic directives to submit jobs with *sbatch*:

```
sbatch <job_script>
```

submits a "job script" to the queue system (see Job directives (section 9.3)).

```
squeue
```

shows all the submitted jobs.

```
scancel <job_id>
```

remove the job from the queue system, canceling the execution of the processes, if they were still running.

9.2 Interactive Sessions

Allocation of an interactive session in the debug partition has to be done through SLURM:

- Interactive session KNL shared, 4 cores:

```
salloc -t 00:10:00 -n 1 -c 4 -J debug -p knl srun --pty /bin/bash
```

- Interactive session KNL exclusive, 64 cores:

```
salloc -t 00:10:00 -n 1 -c 64 -J debug -p knl srun --pty /bin/bash
```

- Interactive session SMP shared, 4 cores:

```
salloc -t 00:10:00 -n 1 -c 4 -J debug -p smp srun --pty /bin/bash
```

- Interactive session SMP exclusive, 32 cores:

```
salloc -t 00:10:00 -n 1 -c 32 -J debug -p smp srun --pty /bin/bash
```

You may add -c <ncpus> to allocate n CPUs.

9.3 Job directives

A job must contain a series of directives to inform the batch system about the characteristics of the job. These directives appear as comments in the job script and have to conform to either the *sbatch* syntaxes.

sbatch syxtax is of the form:

```
#SBATCH --directive=value
```

Additionally, the job script may contain a set of commands to execute. If not, an external script may be provided with the 'executable' directive. Here you may find the most common directives for both syntaxes:

```
# sbatch
#SBATCH --qos=debug
```

This partition is only intended for small tests.

```
# sbatch
#SBATCH --time=HH:MM:SS
```

The limit of wall clock time. This is a mandatory field and you must set it to a value greater than 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 time has passed.

```
# sbatch
#SBATCH -D pathname
```

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.

```
# sbatch
#SBATCH --error=file
```

The name of the file to collect the standard error output (stderr) of the job.

```
# sbatch
#SBATCH --output=file
```

The name of the file to collect the standard output (stdout) of the job.

```
# sbatch
#SBATCH --ntasks=number
```

The number of processes to start.
Optionally, you can specify how many threads each process would open with the directive:

```
# sbatch
#SBATCH --cpus-per-task=number
```

The number of cpus assigned to the job will be the total_tasks number * cpus_per_task number.

```
# sbatch
#SBATCH --ntasks-per-node=number
```

The number of tasks assigned to a node.

```
# sbatch
#SBATCH --reservation=reservation_name
```

The reservation where your jobs will be allocated (assuming that your account has access to that reservation). In some ocasions, node reservations can be granted for executions where only a se t of accounts can run jobs. Useful for courses.

```
#SBATCH --mail-type=[begin/end/all/none]
#SBATCH --mail-user=<your_email>

#Fictional example (notified at the end of the job execution):
#SBATCH --mail-type=end
#SBATCH --mail-user=dannydevito@bsc.es
```

Those two directives are presented as a set because they need to be used at the same time. They will enable e-mail notifications that are triggered when a job starts its execution (begin), ends its execution (end) or both (all). The "none" option doesn't trigger any e-mail, it is the same as not putting the directives. The only requisite is that the e-mail specified is valid **and** also the same one that you use for the HPC User Portal (what is the HPC User Portal, you ask? Excellent question, check it out here!: https://www.bsc.es/user-support/hpc_portal.php)

```
# sbatch
#SBATCH --constraint=<config>
```

To select which Memory Mode do you need, you can choose between "cache", "equal" or "flat". You can check the current configuration of the nodes with the following command:

```
scontrol show node knl[04,06-07,09,11]|grep 'NodeName\|ActiveFeatures'
```

This will print each with its current active features: Cluster Mode (quadrant), Memory mode (cache, equal or flat) and knl because of its architecture. If your desired Memory Mode is not present, please get in contact with support@bsc.es.

```
#SBATCH --x11=[=<all/batch/first/last>]
```

If it is set the job will be handled as graphical and Slurm will assign the necessary resources to the job, so you will be able to execute a graphical command and if you do not close the **current terminal** you will get a graphical window. Sets up X11 forwarding on all, batch host, first or last node(s) of the allocation.

```
# sbatch
#SBATCH --switches=number@timeout
```

By default, Slurm schedules a job in order to use the minimum amount of switches. However, a user can request a specific network topology in order to run his job. Slurm will try to schedule the job for timeout minutes. If it is not possible to request number switches (from 1 to 14) after timeout minutes, Slurm will schedule the job by default.

Variable	Meaning
SLURM_JOBID	Specifies the job ID of the executing job
SLURM_NPROCS	Specifies the total number of processes in the job
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

Examples

sbatch examples

Example for a sequential job:

```
#!/bin/bash
#SBATCH --job-name="test_serial"
#SBATCH -D .
#SBATCH --output=serial_%j.out
#SBATCH --error=serial_%j.err
#SBATCH --ntasks=1
#SBATCH --time=00:02:00
./serial_binary> serial.out
```

The job would be submitted using:

```
> sbatch ptest.cmd
```

Examples for a parallel job:

- Running a hybrid MPI+OpenMP job on one KNL node with 16 MPI tasks, each using 4 CPUs via OpenMP:

```
#!/bin/bash
#SBATCH --job-name=test_parallel
#SBATCH -D .
#SBATCH --output=mpi_%j.out
#SBATCH --error=mpi_%j.err
#SBATCH --ntasks=16
#SBATCH --cpus-per-task=4
#SBATCH --time=00:02:00
#SBATCH --partition=knl
mpirun ./parallel_binary> parallel.output
```

- Running on four KNL nodes with 1 task per node, each using 64 CPUs:

```
#!/bin/bash
#SBATCH --job-name=test_parallel
#SBATCH -D .
#SBATCH --output=mpi_%j.out
#SBATCH --error=mpi_%j.err
#SBATCH --ntasks=4
#SBATCH --cpus-per-task=64
#SBATCH --tasks-per-node=1
#SBATCH --time=00:02:00
#SBATCH --partition=knl
mpirun ./parallel_binary> parallel.output
```

9.4 Interpreting job status and reason codes

When using squeue, Slurm will report back the status of your launched jobs. If they are still waiting to enter execution, they will be followed by the reason. Slurm uses codes to display this information, so in this section we will be covering the meaning of the most relevant ones.

Job state codes

This list contains the usual state codes for jobs that have been submitted:

- **COMPLETED (CD)**: The job has completed the execution.
- **COMPLETING (CG)**: The job is finishing, but some processes are still active.
- **FAILED (F)**: The job terminated with a non-zero exit code.

- **PENDING (PD)**: The job is waiting for resource allocation. The most common state after running "sbatch", it will run eventually.
- **PREEMPTED (PR)**: The job was terminated because of preemption by another job.
- **RUNNING (R)**: The job is allocated and running.
- **SUSPENDED (S)**: A running job has been stopped with its cores released to other jobs.
- **STOPPED (ST)**: A running job has been stopped with its cores retained.

Job reason codes

This list contains the most common reason codes of the jobs that have been submitted and are still not in the running state:

- **Priority**: One or more higher priority jobs is in queue for running. Your job will eventually run.
- **Dependency**: This job is waiting for a dependent job to complete and will run afterwards.
- **Resources**: The job is waiting for resources to become available and will eventually run.
- **InvalidAccount**: The job's account is invalid. Cancel the job and resubmit with correct account.
- **InvaldQoS**: The job's QoS is invalid. Cancel the job and resubmit with correct account.
- **QOSGrpCpuLimit**: All CPUs assigned to your job's specified QoS are in use; job will run eventually.
- **QOSGrpMaxJobsLimit**: Maximum number of jobs for your job's QoS have been met; job will run eventually.
- **QOSGrpNodeLimit**: All nodes assigned to your job's specified QoS are in use; job will run eventually.
- **PartitionCpuLimit**: All CPUs assigned to your job's specified partition are in use; job will run eventually.
- **PartitionMaxJobsLimit**: Maximum number of jobs for your job's partition have been met; job will run eventually.
- **PartitionNodeLimit**: All nodes assigned to your job's specified partition are in use; job will run eventually.
- **AssociationCpuLimit**: All CPUs assigned to your job's specified association are in use; job will run eventually.
- **AssociationMaxJobsLimit**: Maximum number of jobs for your job's association have been met; job will run eventually.
- **AssociationNodeLimit**: All nodes assigned to your job's specified association are in use; job will run eventually.

10 Software Environment

All software and numerical libraries available at the cluster can be found at /apps/. If you need something that is not there please contact us to get it installed (see Getting Help (chapter 11)).

10.1 C Compilers

In the 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.
These are the default sizes of the standard C/C++ datatypes on the machine

Table 1: Default datatype sizes on the machine

Type	Length (bytes)
bool (c++ only)	1
char	1
wchar_t	4
short	2
int	4
long	8
float	4
double	8
long double	16

Distributed Memory Parallelism

To compile MPI programs it is recommended to use the following handy wrappers: mpicc, mpicxx for C and C++ source code. You need to choose the Parallel environment first: module load openmpi /module load impi /module load poe. These wrappers will include all the necessary libraries to build MPI applications without having to specify all the details by hand.

```
% mpicc a.c -o a.exe
% mpicxx a.C -o a.exe
```

10.2 Intel Parallel Studio XE

The Intel Parallel Studio is a package of different tools that allow for advanced profiling, debugging and analyzing of applications, specifically focussed and tuned for Intel processors. Getting Started[6]

- Advisor[7] - Vectorization and Threading

[6] https://software.intel.com/en-us/get-started-with-parallel-studio-xe-for-linux
[7] https://software.intel.com/en-us/get-started-with-advisor

Vector units in the Xeon Phi KNL are 512 bits wide and allow to operate on 16 SP or 8 DP numbers at the same time. Adviser helps to identify which loops are using the full length of these vector registers.

- Inspector XE[8] - Memory and thread debugger.

Use this tool to find races, deadlocks, and illegal memory accesses.

- VTune Amplifier XE[9] - Performance profiler.

Use this tool in the threading and bandwidth optimization stages and for advanced vectorization optimization. Using VTune Guide for Intel Xeon Phi Knights Landing[10]

- Trace Analyzer and Collector[11] - MPI communications performance profiler and correctness checker.

Use this tool in the MPI tuning stage.

Shared Memory Parallelism

OpenMP directives are fully supported by the Intel C and C++ compilers. To use it, the flag -openmp must be added to the compile line.

```
% icc -openmp -o exename filename.c
% icpc -openmp -o exename filename.C
```

You can also mix MPI + OPENMP code using -openmp with the mpi wrappers mentioned above.

Automatic Parallelization

The Intel C and C++ compilers are able to automatically parallelize simple loop constructs, using the option "-parallel" :

```
% icc -parallel a.c
```

10.3 FORTRAN Compilers

In the cluster you can find these compilers :
ifort -> Intel Fortran Compilers

```
% man ifort
```

gfortran -> GNU Compilers for FORTRAN

```
% man gfortran
```

By default, the compilers expect all FORTRAN source files to have the extension ".f", and all FORTRAN source files that require preprocessing to have the extension ".F". The same applies to FORTRAN 90 source files with extensions ".f90" and ".F90".

Distributed Memory Parallelism

In order to use MPI, again you can use the wrappers mpif77 or mpif90 depending on the source code type. You can always man mpif77 to see a detailed list of options to configure the wrappers, ie: change the default compiler.

```
% mpif77 a.f -o a.exe
```

[8] https://software.intel.com/en-us/get-started-with-inspector
[9] https://software.intel.com/en-us/get-started-with-vtune
[10] https://software.intel.com/sites/default/files/managed/1f/eb/Using_Intel_VTune_Amplifier_XE_on_Knights_Landing_1.1.pdf
[11] https://software.intel.com/en-us/get-started-with-itac

Shared Memory Parallelism

OpenMP directives are fully supported by the Intel Fortran compiler when the option "-openmp" is set:

```
% ifort -openmp
```

Automatic Parallelization

The Intel Fortran compiler will attempt to automatically parallelize simple loop constructs using the option "-parallel":

```
% ifort -parallel
```

10.4 Xeon Phi compilation

To produce binaries optimized for the Xeon Phi CPU architecture you should use either Intel compilers or GCC You can load a GCC environment using module:

10.5 Modules Environment

The Environment Modules package (http://modules.sourceforge.net/) provides a dynamic modification of a user's environment via modulefiles. Each modulefile contains the information needed to configure the shell for an application or a compilation. Modules can be loaded and unloaded dynamically, in a clean fashion. All popular shells are supported, including bash, ksh, zsh, sh, csh, tcsh, as well as some scripting languages such as perl.

Installed software packages are divided into five categories:

- Environment: modulefiles dedicated to prepare the environment, for example, get all necessary variables to use openmpi to compile or run programs
- Tools: useful tools which can be used at any time (php, perl, ...)
- Applications: High Performance Computers programs (GROMACS, ...)
- Libraries: Those are tipycally loaded at a compilation time, they load into the environment the correct compiler and linker flags (FFTW, LAPACK, ...)
- Compilers: Compiler suites available for the system (intel, gcc, ...)

Modules tool usage

Modules can be invoked in two ways: by name alone or by name and version. Invoking them by name implies loading the default module version. This is usually the most recent version that has been tested to be stable (recommended) or the only version available.

```
% module load intel
```

Invoking by version loads the version specified of the application. As of this writing, the previous command and the following one load the same module.

```
% module load intel/2017.1
```

The most important commands for modules are these:

- *module list* shows all the loaded modules
- *module avail* shows all the modules the user is able to load
- *module purge* removes all the loaded modules

- *module load <modulename>* loads the necessary environment variables for the selected modulefile (PATH, MANPATH, LD_LIBRARY_PATH...)
- *module unload <modulename>* removes all environment changes made by module load command
- *module switch <oldmodule> <newmodule>* unloads the first module (oldmodule) and loads the second module (newmodule)

You can run "module help" any time to check the command's usage and options or check the module(1) manpage for further information.

11 Getting help

BSC provides users with excellent consulting assistance. User support consultants are available during normal business hours, Monday to Friday, 09 a.m. to 18 p.m. (CEST time).

User questions and support are handled at: support@bsc.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 files. Please contact BSC if you have any questions or comments regarding policies or procedures.

Our address is:

```
Barcelona Supercomputing Center - Centro Nacional de Supercomputación
C/ Jordi Girona, 31, Edificio Capilla 08034 Barcelona
```

12 Frequently Asked Questions (FAQ)

You can check the answers to most common questions at BSC's Support Knowledge Center[12]. There you will find online and updated versions of our documentation, including this guide, and a listing with deeper answers to the most common questions we receive as well as advanced specific questions unfit for a general-purpose user guide.

13 Appendices

13.1 SSH

SSH is a program that enables secure logins over an insecure network. It encrypts all the data passing both ways, so that if it is intercepted it cannot be read. It also replaces the old an insecure tools like telnet, rlogin, rcp, ftp,etc. SSH is a client-server software. Both machines must have ssh installed for it to work.

We have already installed a ssh server in our machines. You must have installed an ssh client in your local machine. SSH is available without charge for almost all versions of UNIX (including Linux and MacOS X). For UNIX and derivatives, we recommend using the OpenSSH client, downloadable from http://www.openssh.org, and for Windows users we recommend using Putty, a free SSH client that can be downloaded from http://www.putty.org. Otherwise, any client compatible with SSH version 2 can be used. If you want to try a simpler client with multi-tab capabilities, we also recommend using Solar-PuTTY (https://www.solarwinds.com/free-tools/solar-putty).

This section describes installing, configuring and using PuTTy on Windows machines, as it is the most known Windows SSH client. No matter your client, you will need to specify the following information:

- Select SSH as default protocol
- Select port 22
- Specify the remote machine and username

For example with putty client:

[12] https://www.bsc.es/user-support/faq.php

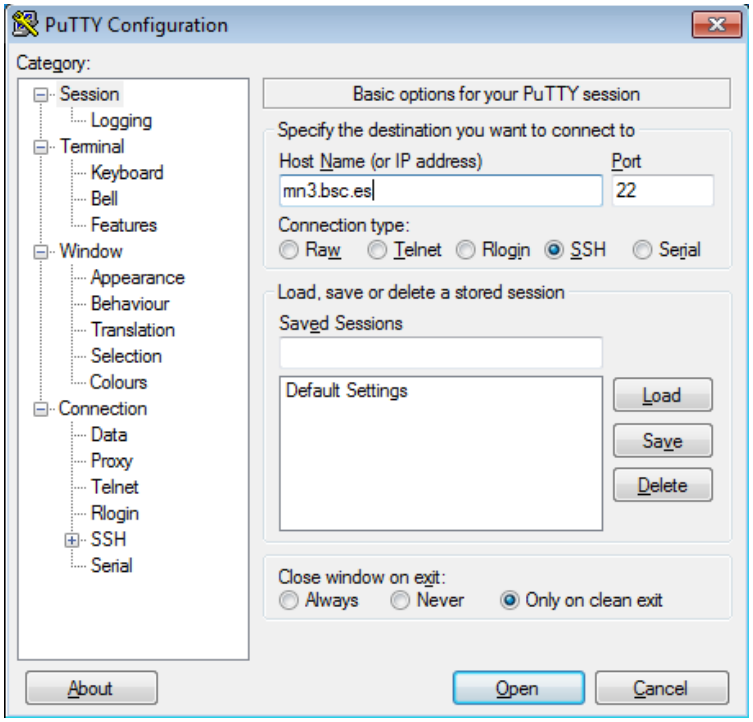

Figure 1: Putty client

This is the first window that you will see at putty startup. Once finished, press the **Open** button. If it is your first connection to the machine, your will get a *Warning* telling you that the host key from the server is unknown, and will ask you if you are agree to cache the new host key, press Yes.

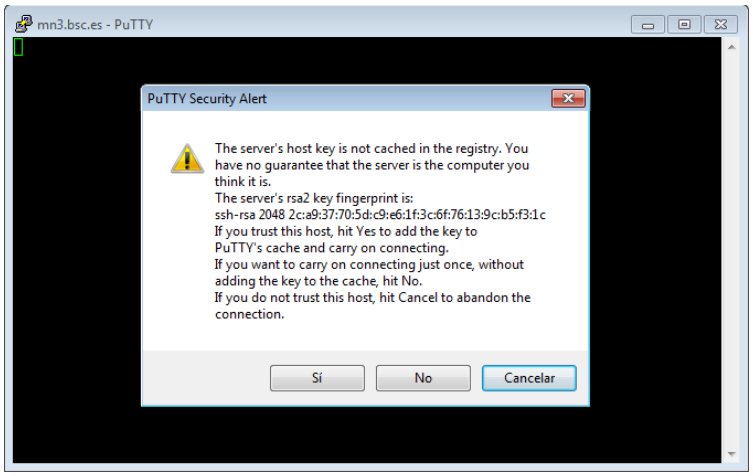

Figure 2: Putty certificate security alert

IMPORTANT: If you see this warning another time and you haven't modified or reinstalled the ssh client, please do *not* log in, and contact us as soon as possible (see Getting Help (chapter 11)).

Finally, a new window will appear asking for your login and password:

Generating SSH keys with PuTTY

First of all, open PuTTY Key Generator. You should select Type RSA and 2048 or 4096 bits, then hit the "Generate" button.

After that, you will have to move the mouse pointer inside the blue rectangle, as in picture:

You will find and output similar to the following picture when completed

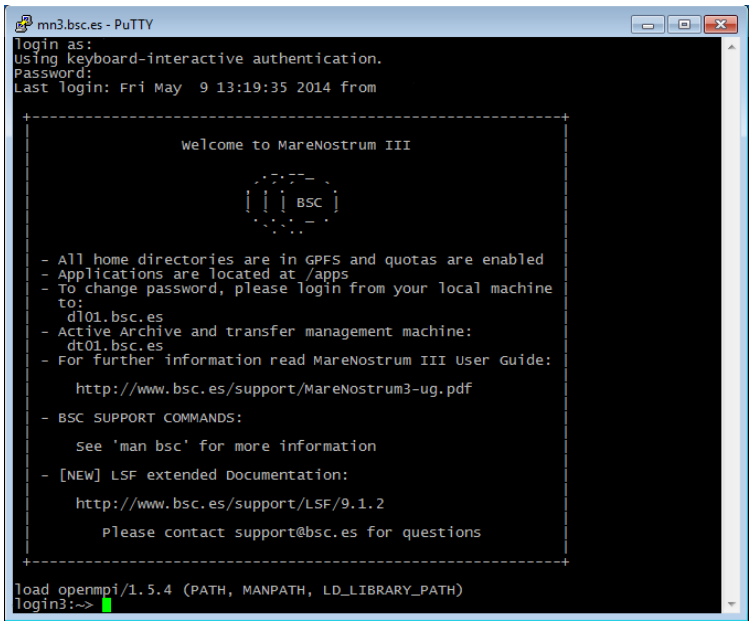

Figure 3: Cluster login

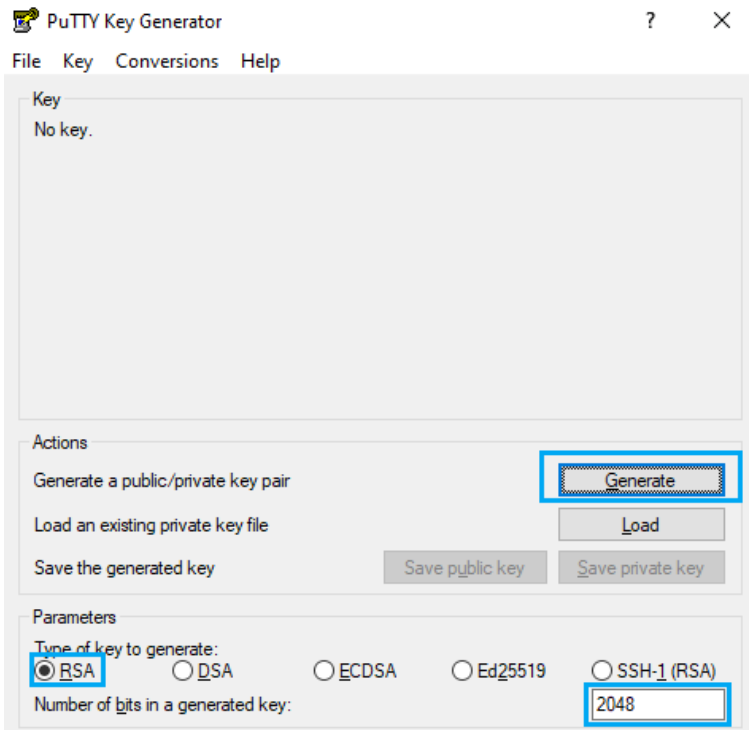

Figure 4: Public key PuTTY window selection

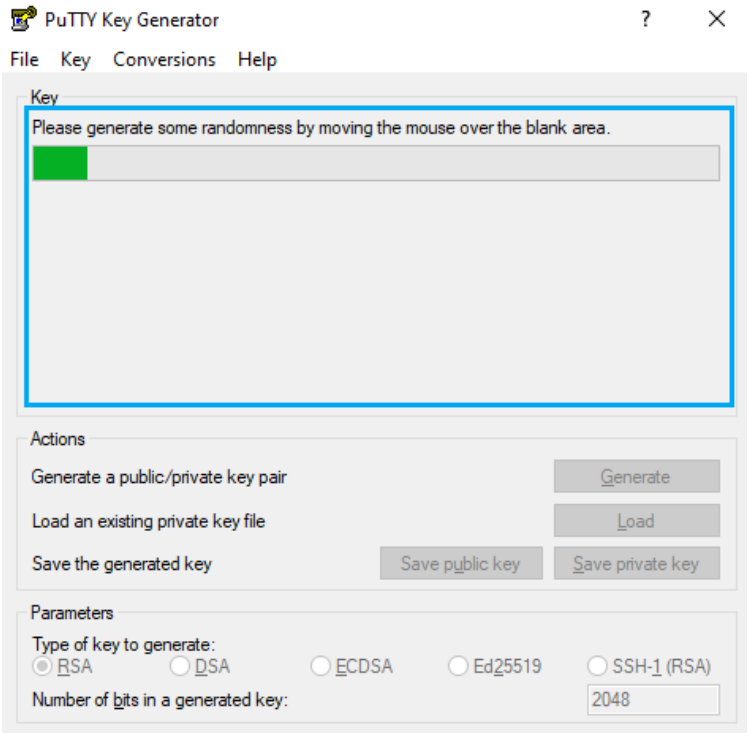

Figure 5: PuTTY box where you have to move your mouse

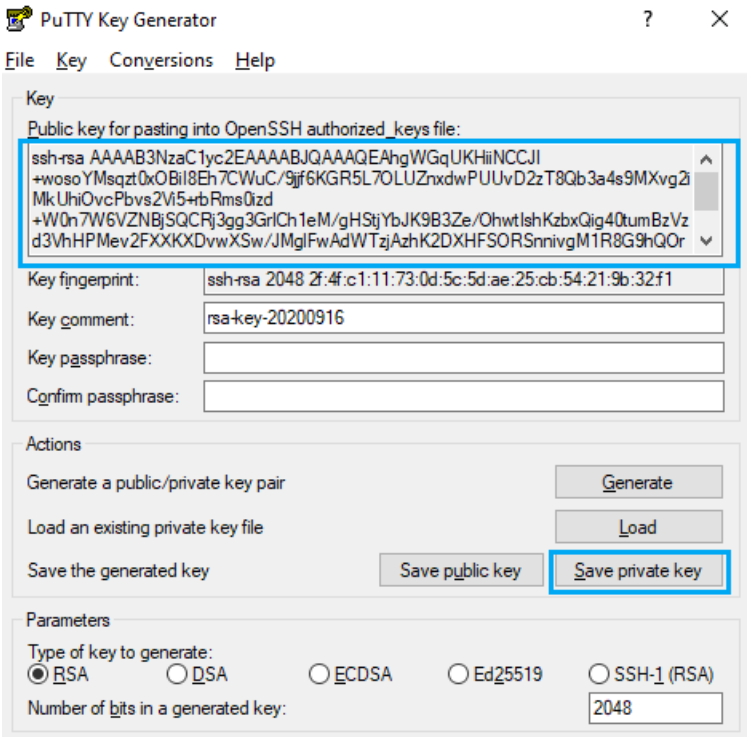

Figure 6: PuTTY dialog when completed

This is your public key, you can copy the text in the upper text box to the notepad and save the file. On the other hand, click on "Save private key" as in the previous picture, then export this file to your desired path.

You can close PuTTY Key Generator and open PuTTY by this time,

To use your recently saved private key go to Connection -> SSH -> Auth, click on Browse... and select the file.

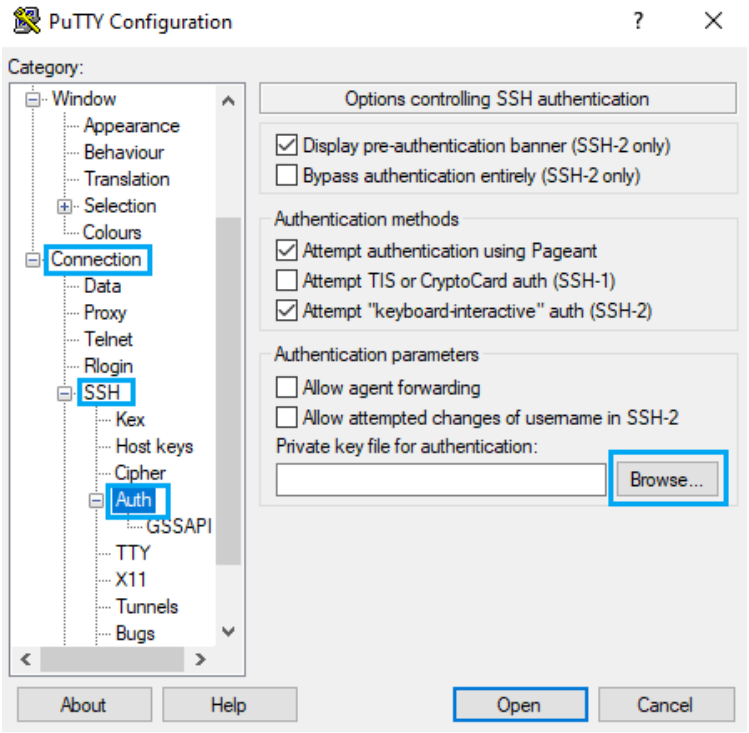

Figure 7: PuTTY SSH private key selection

13.2 Transferring files on Windows

To transfer files to or from the cluster you need a secure FTP (SFTP) o secure copy (SCP) client. There are several different clients, but as previously mentioned, we recommend using the Putty clients for transferring files: **psftp** and **pscp**. You can find them at the same web page as PuTTY (http://www.putty.org[13]), you just have to go to the download page for PuTTY and you will see them in the "alternative binary files" section of the page. They will most likely be included in the general PuTTY installer too.

Some other possible tools for users requiring graphical file transfers could be:

- WinSCP: Freeware SCP and SFTP client for Windows (http://www.winscp.net)
- Solar-PuTTY: Free alternative to PuTTY that also has graphical interfaces for SCP/SFTP. (https://www.solarwinds.com/free-tools/solar-putty)

Using PSFTP

You will need a command window to execute psftp (press start button, click run and type cmd). The program first asks for the machine name (mn1.bsc.es), and then for the username and password. Once you are connected, it's like a Unix command line.

With command **help** you will obtain a list of all possible commands. But the most useful are:

- get file_name : To transfer from the cluster to your local machine.
- put file_name : To transfer a file from your local machine to the cluster.

[13] http://www.putty.org/

- cd directory : To change remote working directory.
- dir : To list contents of a remote directory.
- lcd directory : To change local working directory.
- !dir : To list contents of a local directory.

You will be able to copy files from your local machine to the cluster, and from the cluster to your local machine. The syntax is the same that cp command except that for remote files you need to specify the remote machine:

```
Copy a file from the cluster:
> pscp.exe username@mn1.bsc.es:remote_file local_file
Copy a file to the cluster:
> pscp.exe local_file username@mn1.bsc.es:remote_file
```

13.3 Using X11

In order to start remote X applications you need and X-Server running in your local machine. Here are two of the most common X-servers for Windows:

- Cygwin/X: http://x.cygwin.com
- X-Win32 : http://www.starnet.com

The only Open Source X-server listed here is Cygwin/X, you need to pay for the other.
Once the X-Server is running run putty with X11 forwarding enabled:

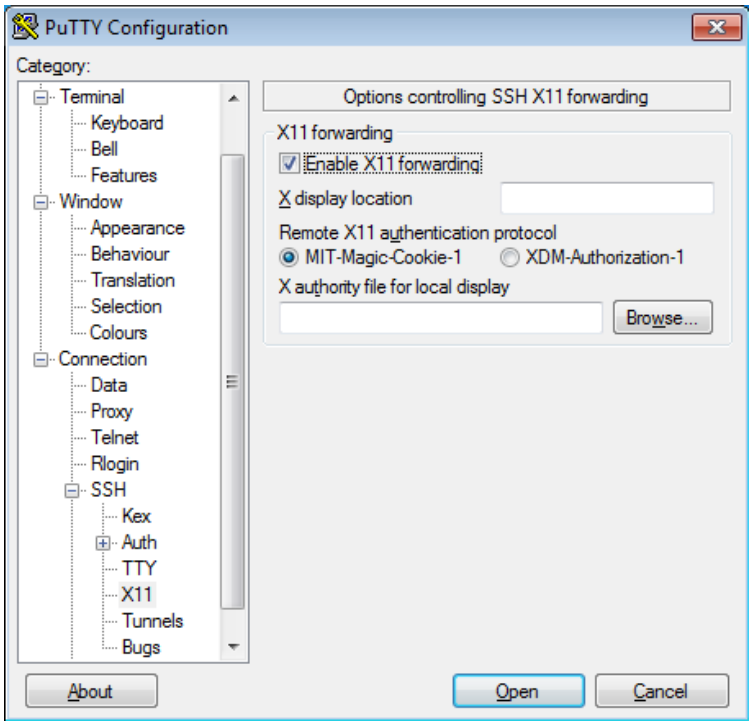

Figure 8: Putty X11 configuration

I tried running a X11 graphical application and got a GLX error, what can I do?

If you are running on a macOS/Linux system and, when you try to use some kind of graphical interface through remote SSH X11 remote forwarding, you get an error similar to this:

```
X Error of failed request: BadValue (integer parameter out of range for operation)
  Major opcode of failed request: 154 (GLX)
  Minor opcode of failed request: 3 (X_GLXCreateContext)
  Value in failed request: 0x0
  Serial number of failed request: 61
  Current serial number in output stream: 62
```

Try to do this fix:
macOS:

- Open a command shell, type, and execute:

```
$ defaults write org.macosforge.xquartz.X11 enable_iglx -bool true
```

- Reboot your computer.

Linux:

- Edit (as root) your Xorg config file and add this:

```
Section "ServerFlags"
    Option "AllowIndirectGLX" "on"
    Option "IndirectGLX" "on"
EndSection
```

- Reboot your computer.

This solves the error most of the time. The error is related to the fact that some OS versions have disabled indirect GLX by default, or disabled it at some point during an OS update.

13.4 Requesting and installing a .X509 user certificate

If you are a BSC employee (and you also have a PRACE account), you may be interested in obtaining and configuring a x.509 Grid certificate. If that is the case, you should follow this guide. First, you should obtain a certificate following the details of this guide (you must be logged in the BSC intranet):

- https://intranet.bsc.es/help-and-support/operations-services/personal-digital-certificates

Once you have finished requesting the certificate, you must download it in a ".p12" format. This procedure may be different depending on which browser you are using. For example, if you are using Mozilla Firefox, you should be able to do it following these steps:

- Go to "Preferences".
- Navigate to the "Privacy & Security" tab.
- Scroll down until you reach the "Certificates" section. Then, click on "View Certificates..."
- You should be able to select the certificate you generated earlier. Click on "Backup...".
- Save the certificate as "usercert.p12". Give it a password of your choice.

Once you have obtained the copy of your certificate, you must set up your environment in your HPC account. To acomplish that, follow these steps:

- Connect to dt02.bsc.es using your PRACE account.
- Go to the GPFS home directory of your HPC account and create a directory named ".globus".
- Upload the .p12 certificate you created earlier inside that directory.

- Once you are logged in, insert the following commands (insert the password you chose when needed):

```
module load prace globus
cd ~/.globus
openssl pkcs12 -nocerts -in usercert.p12 -out userkey.pem
chmod 0400 userkey.pem
openssl pkcs12 -clcerts -nokeys -in usercert.p12 -out usercert.pem
chmod 0444 usercert.pem
```

Once you have finished all the steps, your personal certificate should be fully installed.