Exascale Computing Research OB1
cluster access documentation

This document describes the different steps you need to follow to access the **OB1** cluster.

READ THIS DOCUMENT FULLY AND THOROUGHLY BEFORE ATTEMPTING THE PROCEDURES DESCRIBED BELOW

0) Generating authentication RSA key pairs:

If you do not already have ssh keys generated on your workstation, you should generate a pair using the following command:

```
user@laptop$ ssh-keygen
```

This command will create an RSA key pair in **~/.ssh/id_rsa.pub** (**public** key) and **~/.ssh/id_rsa** (**private** key) that will allow you to authenticate without using passwords.

1) Accessing the gateway node:

You need to login to **gob1** (gateway OB1) first with the provided initial credentials to renew your password and copy your public ssh key. To do so, you need to add the following lines to your **~/.ssh/config** on your local workstation:

```
Host gob1
User USERNAME
Hostname 193.55.217.13

Host fob1
User USERNAME
ProxyCommand ssh gob1 ncat %h %p
```

Make sure to replace **USERNAME** by the given username. Then, you can connect using the following command:

```
user@laptop$ ssh gob1
```

You'll be asked to change your password at first. Once your password is changed successfully, you should copy your public ssh key into **gob1** using the following command:

```
user@laptop$ ssh-copy-id -i ~/.ssh/id_rsa.pub gob1
```

After authentication has succeed, you will be able to access **gob1** without having to enter any authentication tokens. The following command

```
user@laptop$ ssh gob1
```

should connect you to the **gob1** gateway node automatically.

2) **Accessing the frontal login node:**

Once your access to **gob1** is set, you need to set up access to the frontal node **fob1** that can allow you to access the compute nodes. First, connect to **fob1 and** change your password.

```
user@laptop$ ssh fob1
```

Once your password is successfully changed, copy your public ssh key into **fob1**.

```
user@laptop$ ssh-copy-id -i ~/.ssh/id_rsa.pub fob1
```

After the authenticaction is complete, connect to **fob1** to test your access**:**

```
user@laptop$ ssh fob1
```

Important: From now on, you do not need to login to **gob1** directly to access **fob1**. You can access **fob1** from your laptop or home workstation directly with **gob1** playing the role of a transparent gateway as set in the **~/.ssh/config** file. You should exit back to your local workstation terminal and try to login to **fob1**.

3) Setting up accesses to the compute nodes:

There are 10 large compute nodes available on the OB1 cluster. 5 KNL (Intel Xeon Phi - Knights Landing) nodes, and 4 HSW (Haswell) nodes. The KNL nodes **knl01**, **knl03**, and **knl05** have 256 active cores/threads and DRAM ranging from 96GB to 110GB. The **knl06** node is equipped with a 288 cores/threads CPU and 110GB of DRAM. The nodes **knl02**, **knl03**, **knl05**, **knl06** all have an additional 16GB of MCDRAM (HBM – High Bandwidth Memory) visible as NUMA node 1. **knl07** only has HBM.

Once you are connected to **fob1,** you need to generate an RSA key pair (again) that will be used to authenticate your access to the compute nodes.

```
userXXXX@fob1$ ssh-keygen
```

Then, you need to update your password on all the compute nodes you want to use:

```
userXXXX@fob1$ ssh knl01 → change password
userXXXX@fob1$ ssh knl02 → change password
…
```

Once the keys generated and the password changed successfully, you need to copy the public key into the compute node **knl01**.

```
userXXXX@fob1$ ssh-copy-id -i ~/.ssh/id_rsa.pub knl01
```

Once you are authenticated and the keys successfully copied, you can login to any compute node seemlessly without using a password.

4) Tools available:

A fairly uniform set of tools is already installed on the login and compute nodes.

0) MPICH
1) gcc, g++, gfortran
2) make
3) cmake
4) numactl
5) git
6) BLAS
7) perf

5) Useful commands

- For copying files from your laptop/workstation to the cluster's file system, you can use the following commands:

1- For copying files:

```
user@laptop$ scp PATH_TO_FILE fob1:
```

2- For copying directories:

```
user@laptop$ scp -r PATH_TO_DIRECTORY fob1:
```

The target file will be copied automatically to your user home directory on the cluster's filesystem.

You can also use **sshfs** to remotely mount your home directory (or any other directory) on your laptop/workstation's filesystem using the following command:

```
user@laptop$ sshfs fob1:PATH_TO_DIRECTORY PATH_TO_LOCAL_MOUNT_POINT
```

On the KNL compute nodes, two modules are available: the **Intel OneAPI** toolset, and the **MAQAO** performance profiling and analysis tool. You also have native access to **numactl** and **taskset** for controlling process/thread placement on CPU cores, **perf** for performance profiling, and **cpupower** to verifying the CPU cores frequency states.

- For listing the available modules and packages, you can use to following command:

```
userXXXX@knlYY$ module avail
```

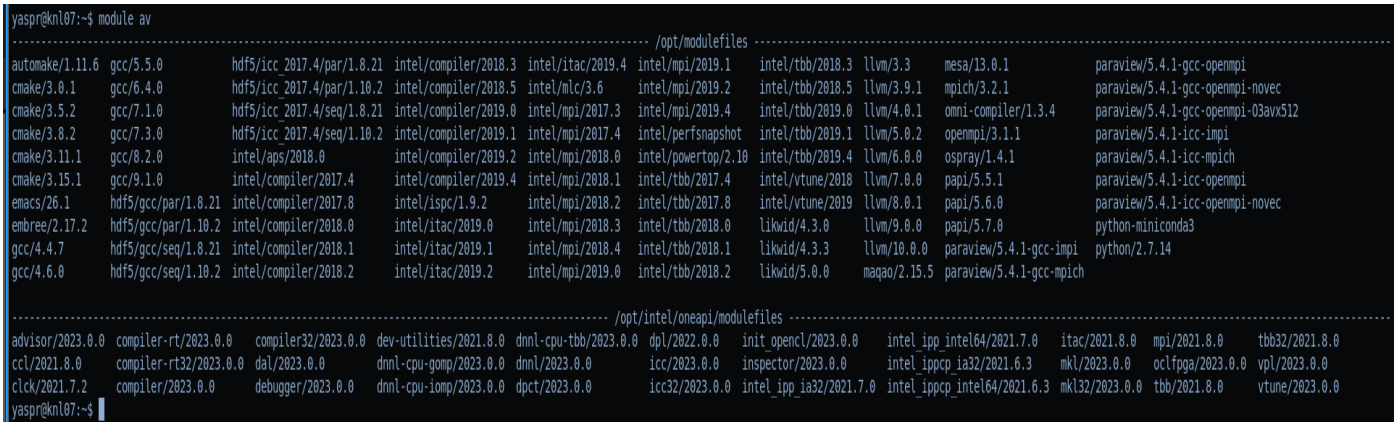

Image1 – List of available modules on the compute nodes

- For loading the a module's package, for example the **Intel OneAPI** compiler and libraries, you can use the following command:

```
userXXXX@knlYY$ module load compiler/latest mkl/latest
```

After loading the **Intel OneAPI** environment, you can use the **icc** and **icx** compilers, as well as the **MKL** library.

- For loading the **MAQAO** module, you can use the following command:

```
userXXXX@knlYY$ module load 2.15.0
```

For more help about **MAQAO**, you can use the following commands:

```
userXXXX@knlYY$ maqao –help
userXXXX@knlYY$ maqao oneview –help
```

If you encounter any issues, feel free to contact me at: yaspr@liparad.uvsq.fr

Enjoy ;)

Yaspr, the NSA (Network Security Administrator)