Chair for Computer Science 12 - High Performance Computing ***www.hpc.rwth-aachen.de***
Müller, Cramer, Miller, Schwitanski ***contact@hpc.rwth-aachen.de***

High-Performance Computing, Winter 2021

Exercise 0: RWTH Cluster Environment

Task 1. Get familiar with the RWTH Compute Cluster

The IT Center of the RWTH Aachen University has been operating a UNIX cluster since 1994 and supporting Linux since 2004 and Windows since 2005.
The cluster is operated to serve the computational needs of researchers from RWTH Aachen University, the Jülich Research Center and other universities in North-Rhine-Westphalia. This means that every employee of one of these universities may use the cluster for research purposes. Furthermore, students of RWTH Aachen University can get an account in order to become acquainted with parallel computers and learn how to program them.

Task 1.1. Create an HPC Account
If not already done, create an HPC account at `http://www.rwth-aachen.de/selfservice` by activating "Hochleistungsrechnen RWTH Aachen" in the menu "Accounts and Passwords / Create Account" ("Accounts und Passwörter / Accounts anlegen").
Afterwards, send your RWTH user ID to `contact@hpc.rwth-aachen.de` with the subject "[HPC21] (RWTH User ID)" (e.g., "[HPC21] ab123456"). Then, we will add you to a certain project that enables some more rights on our cluster (see below). Be aware that this will take up to one day!

Task 1.2. Connect to the RWTH Network
You have to be connected to the network of RWTH Aachen University to get access to the RWTH Compute Cluster. Most simple ways are *eduroam* or *VPN*. Please find instructions how to connect to the VPN in the IT Center Help: `https://help.itc.rwth-aachen.de/service/vbf6fx0gom76`

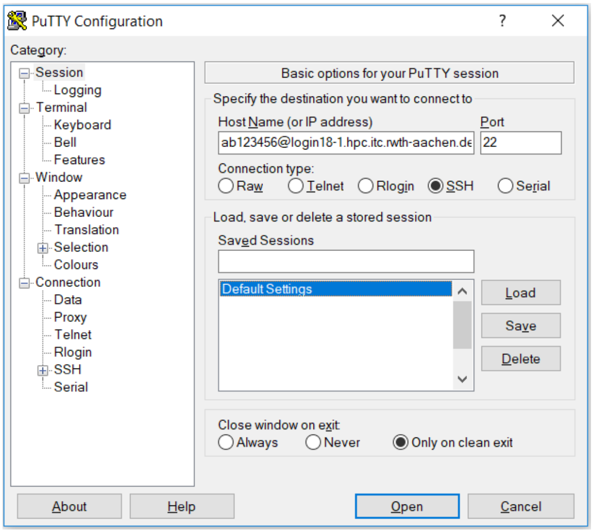
Task 1.3. Login to a Frontend Node
There are multiple ways to create a connection to one of the frontend nodes of the RWTH Compute Cluster (`https://help.itc.rwth-aachen.de/service/rhr4fjjutttf/article/cd350ff0de1f460bac545b6cde32e8a7`):

1. FastX: We provide dedicated dialog systems (e.g., login18-x-1.hpc.itc.rwth-aachen.de or login18-x-2.hpc.itc.rwth-aachen.de) for remote desktop sessions. FastX allows to run remote X11 (graphical) sessions even across low-bandwidth network connections, as well as reconnecting to running sessions. For students that did not use SSH so far, we recommend this solution. In particular, using the browser-based client should work on most systems without installing any software. More information on the installation and setup can be found in the IT Center Help: `https://help.itc.rwth-aachen.de/service/rhr4fjjutttf/article/25f576374f984c888bb2a01487fef193`

2. SSH from Linux or macOS: Users can use a ssh connection out of a terminal:
   `$ ssh -Y`[1] *`<RWTH user ID>`*`@login18-1.hpc.itc.rwth-aachen.de`

3. SSH from Windows – use PuTTY: Download PuTTY from `http://www.putty.org/`, extract or install and configure it to connect to *login18-1* or *login18-2* (see Fig. 1).

[1] -Y Enables trusted X11 forwarding



Host Name (or IP address): ab123456@login18-1.hpc.itc.rwth-aachen.de
Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 1: PuTTY configuration

Task 1.4. Download Exercise Archive
You have three options to get the exercise archive file `exercise0.tar.gz` to the frontend node into your `$HOME` directory (Note: The `$HOME` directory is the directory `/home/xy123456/` where `xy123456` is your RWTH User ID.):

1. If you use FastX, then just use Firefox or another browser on the graphical frontend node to download the archive file from Moodle and copy it to your `$HOME` directory.

2. If you use SSH on a Linux or macOS system, then download the archive file from Moodle to your local machine and copy it using `scp` to your `$HOME` directory:
   ```
   $ scp exercise0.tar.gz xy123456@login18-1.hpc.itc.rwth-aachen.de:
   ```

3. If you have sent your RWTH User ID to `contact@hpc.rwth-aachen.de` and you have been added to the corresponding group, then you can also copy the file from the directory `/home/lect0067` to your `$HOME` directory:
   ```
   $ cp /home/lect0067/exercise0.tar.gz $HOME
   ```

Extract the tarball in your current directory (FastX users have to start a terminal session on the desktop to run commands)

```
$ tar -xvf exercise0.tar.gz
```

The created folder contains subfolders for each task.

Task 1.5. Batch Job
The frontend nodes of the RWTH Compute Cluster are just meant for login, compiling and short (!) tests. For usual tests and performance measurements, you should always use the batch environment. That means that you submit a "batch job" that contains a description of what you want to do and which is scheduled as soon as there are corresponding resources available. Please note that depending on your requested resources, you will probably have to wait shorter or longer. So, try to request minimal values in runtime, memory and core number.

If you have sent your RWTH User ID to `contact@hpc.rwth-aachen.de`, you have the possibility to submit to a special lecture project that reserves some compute resources for you for the purpose of completing your HPC exercises. Note that these compute resources are for all students in this lecture. Please use them responsibly!

A short example of a serial batch job may look like this (you can find it in `batchLect0067.sh`):

```
#!/bin/zsh

### Job name
#SBATCH --job-name=hello-world

### File / path where STDOUT & STDERR will be written
### %J is the job id
#SBATCH --output=output_%J.txt

### Request project lecture
### Runtime limit: 20 min
### Max 2 jobs from one user can run simultaneously
#SBATCH -A lect0067

### Request the time you need for execution in minutes
### Format hours:min:sec
#SBATCH --time=00:05:00

### Request vitual memory. M is the default and can therefore be omitted,
### but could also be K(ilo)|G(iga)|T(era)
#SBATCH --mem-per-cpu=512M

### Specify your mail address
###SBATCH --mail-user=<specify_your_mail>

### Send a mail when job is done
###SBATCH --mail-type=END

helloWorld.out
```

Test this batch script by running the following commands in a terminal:

First compile the `helloWorld.c` source file by
`$ $CC -o helloWorld.out helloWorld.c`
Then execute the batch script on the frontend node:
`$ ./batchlect0067.sh`
If that is not working, make sure that this file has execution rights for the user. You can set these by
`$ chmod u+x ./batchLect0067.sh`
Executing the batch script on the frontend is always a good testing method.
Now, submit the batch job to the batch environment by
`$ sbatch batchLect0067.sh`
With `squeue -u ‘whoami‘`, you can check whether your job is in the batch queue and whether it is pending or running.
Does the generated output file look like expected?
Once the job was executed you can determine the status with:
`$ sacct`
Was the execution successful? On which partition it was executed?

Discussion of this exercise on November 5, 2021.