High performance scientific computing in C++

HPC C++ Course 2023

29 May – 02 June 2023 | Sandipan Mohanty | Forschungszentrum Jülich, Germany

High performance scientific computing in C++

Supercomputer access for exercises

- Please login to the Jupyter-JSC system
- After logging in, try to add a new jupyterlab. Choose JUSUF as the system and training2316 as the project.
- For the partition choose **LoginNode**, and then start. Wait until the swirly things stop and you see the panel.
- What we will most need from there is the terminal, which should be at the bottom.
- In the terminal type this:

```
$ source $PROJECT/local/setup.sh
```

- After this, your paths should be set correctly. Test it using

```
$ g++ --version
$ clang++ --version
```

You should see GCC version 13.1 and Clang version 16.0.

Direct SSH connection to JUSUF

- Follow the instructions in the page https://apps.fz-juelich.de/jsc/hps/jusuf/cluster/access.html#ssh-login to generate a suitable SSH key for JUSUF. Certain key types are preferred, and the recommended key type is ed25519.

```
$ ssh-keygen -a 100 -t ed25519 -f ~/.ssh/id_ed25519_jsc
$
```

- Upload the generated **public key** to JuDoor
  - Login to JuDoor
  - Click on "Manage SSH-keys" next to JUSUF
  - Select your ed25519 public key file using the "Browse" button. This file has a ".pub" extension.
  - Pay special attention to the "from clause". This goes in the box to the right of the public key filename box. If you are in a hurry, use your current IP address, which is displayed in the colour coded field at the top in the section "Upload your public keys"
  - After filling in a suitable "from clause", click on the Start upload of SSH-keys button
- Wait

Direct SSH connection to JUSUF

- Open SSH session on JUSUF using your ed25519 key like this:

```
$  ssh -i ~/.ssh/id_ed25519_jsc <yourid>@jusuf.fz-juelich.de
```

- ... or, make it a little easier for yourself:
  - Add a section for JUSUF in your SSH config file `~/.ssh/config`

```
1  Host jusuf jusuf??
2    Hostname %h.fz-juelich.de
3    Match Host jusuf.fz-juelich.de,jusuf??.fz-juelich.de
4    User yourid
5    IdentityFile ~/.ssh/id_ed25519_jsc
6    ServerAliveInterval 60
```

  - Open SSH session like this: `$ ssh jusuf`
- After opening a new SSH session, set your project to training2316 and run the setup script:

```
$ jutil env activate -p training2316
$ source $PROJECT/local/setup.sh
$
```

High performance scientific computing in C++

- The setup script must be run at the beginning of every new login to JUSUF for this course.
- It creates user specific working directories, downloads and updates course material and sets up the environment variables for compilers and libraries.
- After the script `setup.sh` is sourced, the following environment variables (EV) and additional shortcuts (SC) are available
  - `cxx2023`: (EV) Location of your private working area for the course
  - `swhome`: (EV) Top level folder for software installations for compilers and libraries
  - `cdp`: (SC) Change directory to the top level of your private workspace
  - `pathadd`: (SC) Prepend a new folder to PATH. E.g., `pathadd /x/y/z/bin`
  - `pathrm`: (SC) Remove a folder from PATH
  - `libpathadd`, `libpathrm`: (SC) Same as above, but for `LD_LIBRARY_PATH`, `LD_RUN_PATH`, `LIBRARY_PATH`
  - `incpathadd`, `incpathrm`: (SC) Same, but for `CPATH`, which is searched by the compilers for include files.
  - `cmpathadd`, `cmpathrm`: Same, but for `CMAKE_PREFIX_PATH`
  - `G`: (SC) Alias for `g++` using common options `-std=c++20 -pedantic -Wall -O3 -march=native`
  - `C`: (SC) Similar to `G`, but for Clang. It also uses Clang's own implementation of the standard library, `libc++`

High performance scientific computing in C++

- The folder `yourworkspace/software`: Any software you build and install with this installation prefix will be found by the compilers and CMake
- Run simple compilation and small programs on the Login node, as you would on your laptop.
- For heavier workloads, we will use the batch system during the course. Run the executable `a.out` using 64 maximum threads on a JUSUF compute node as follows:
  `batch_run --cpus-per-task=64 a.out [OPTIONS]` The alias `batch_run` will be updated for each course day to always use the reservations made on the supercomputer for that course day.
- The path manipulation utilities used in the course are available with the course material in the file `code/bash/pathutils.sh`. It contains only BASH functions like `pathadd`, and nothing specific to our setup on JUSUF. Similarly, the aliases `G`, `C` can be found in `code/bash/aliases.sh`.
- The contents of these first few slides, since you may need to look them up later, are placed in the file `utilities.pdf`