Course Information

Parallel and Distributed Programming

Maya Neytcheva, Marina Shimchenko, Tobias Lass
maya.neytcheva@it.uu.se, marina.shimchenko@it.uu.se, tobias.lass.9583@student.uu.se

Computer resources for using the High Performance Message Passing Library implementation OpenMPI

Own laptop

You can use your own computer for code development and initial testing. OpenMPI is readily available for both Linux and Mac. On Windows, the best solution might be to install Windows Subsystem for Linux (WSL) and then follow Linux instructions, but this is not trivial. If you have a package manager (e.g. brew, apt, yum, rpm or conda) that you know how to use, we recommend that you use it for installation of openmpi, e.g., `brew install open-mpi`, `apt install openmpi-bin`.

If not, you may follow the instructions below to compile it from source. If you don't intend to compile and run MPI programs on you own computer, you don't need to install it. The following instructions will help you installing OpenMPI on your machine. It may take about 20-25 min or less, depending on your computer.

1. Create a temporary directory for compiling OpenMPI. You can do this in a terminal by typing `mkdir -p $HOME/local/src`

2. Download `openmpi-4.1.4.tar.gz` from `http://www.open-mpi.org/software/ompi/v4.1`

3. Move the file `openmpi-4.1.4.tar.gz` to the directory just created:
   `mv $HOME/Downloads/openmpi-4.1.4.tar.gz $HOME/local/src/`

4. Change to the directory and extract the package using

   ```
   cd $HOME/local/src
   tar -xzf openmpi-4.1.4.tar.gz
   ```

5. Go into the source directory `cd openmpi-4.1.4`

6. Configure, compile and install by executing the following commands

   ```
   ./configure --prefix=$HOME/opt/openmpi
   make all
   make install
   ```

   This will install OpenMPI in your home directory in the sub-folder `opt/openmpi`. You can speed up the compilation by replacing the make all command with `make -j4 all` (this will compile using 4 cores).

7. Remove the temporary directories:

```
cd
rm $HOME/local/src/openmpi-4.1.4.tar.gz
rm -r $HOME/local/src/openmpi-4.1.4
```

To use MPI you will have to adapt your `PATH` and `LD_LIBRARY_PATH` environment variable:

```
echo "export PATH=\$PATH:\$HOME/opt/openmpi/bin" >> $HOME/.bashrc
echo "export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:\$HOME/opt/openmpi/lib" \
>> $HOME/.bashrc
source $HOME/.bashrc
```

This appends the two lines to your `.bashrc` file which is executed when starting a terminal session, and reexecutes it.

To compile your `MPI C` programs you have to use `mpicc` with the same arguments as you would use for `gcc`. To run a program PROG with N MPI processes, you would then use `mpirun -np N PROG`. To uninstall OpenMPI just delete the folder `opt/openmpi` in your home directory and remove the corresponding two lines from the `.bashrc` file in your home directory. You can find more information on `http://www.open-mpi.org`

8. To test the functionality of your installation, you can try the instructions and the test codes from Lab 1.

IT Linux hosts

OpenMPI is available on the Linux hosts, listed at `https://www.it.uu.se/datordrift/maskinpark/linux`.

Options to connect to the Linux servers (`(gullviva, tussilago and vitsippa).it.uu.se`:

- From a Linux laptop you can do directly (`gullviva` is as an example):

  `ssh -Y` *`your-ACA-username`*`@gullviva.it.uu.se`

- From a Windows laptop you have to use e.g. MobaXTerm to use the above ssh command
- In the Windows-based lab halls, after you have logged in, you follow the steps:
  1. From Software Center (ZenWorks) start `ThinLinc`.
  2. In the window that opens, choose as a server `thinlinc.student.it.uu.se`, login with your ACA username and password.
  3. In the window which opens, click (`'Applications'` in the top-left corner. Choose `'System tools' -> 'Terminal'`.
  4. In the terminal window make an (`ssh` connection to the Linux servers, as above.
- From a Mac you have to install xQuartz, restart your machine and enter the above command into something that uses the same environment as your terminal.

The `-Y` flag is used to run programs with a GUI. If you are not running a GUI program you can ssh without the `-Y` flag and do not have to follow these setups.

The cluster Snowy at UPPMAX

Throughout the course, we can use the Snowy compute cluster at UPPMAX `https://uppmax.uu.se/resources/systems/the-snowy-cluster/`.

To get access, you need to have a login account. For obtaining an account on UPPMAX, follow the steps below.

1. Sign up for a SNIC account at `https://supr.snic.se/`. Click "Register New Person", and then "Register Via SWAMID" to register with your existing UU login.
2. Login with your new SNIC account, and request membership in the course project. Click on "Projects" in the left menu bar, search for project "**uppmax2023-2-13**" in the search field, and then click request. We will check your course registration status, and approve the request.
3. Once approved, sign up for an UPPMAX login account by clicking on "Accounts" in the left menu bar, and then following the instructions.

Once you have received an email confirming the creation of your UPPMAX account, verify that you can login by issuing the following `ssh` command.

```
ssh -Y your-new-username@rackham.uppmax.uu.se
```

If it does not work immediately, please wait a few minutes and try again since there might be some setup delay. As stated in the UPPMAX account email, the first thing you should do is change your password which is done using the `passwd` command.

SLURM

The UPPMAX systems use the batch system SLURM to control the allocation of resources and submission of compute jobs. Although UPPMAX has been quite generous with resources, we do have a limited quota of core hours which all students will share. Make sure to limit your runs both in time and in the number of cores you allocate, as to not prevent other students from working. Here, there is no strict enforcement preventing overuse; rather, a "freedom with responsibility" approach is used. Deliberate misuse will be detected and can result in suspension from the computer system.

For more information on SLURM and how to use the UPPMAX clusters, see the second lab instruction, and the following useful links:

- `https://www.uppmax.uu.se/support/user-guides/guide--first-login-to-uppmax/`
- `https://www.uppmax.uu.se/support/user-guides/slurm-user-guide/`
- `https://www.uppmax.uu.se/support/user-guides/snowy-user-guide/`

Parallel debugger `Allinea DDT`

The debugger is available at Snowy. See the user guide at `https://www.uppmax.uu.se/support/user-guides/allinea-ddt-user-guide/`. However, since Snowy does not have login nodes, to start `Allinea DDT` on it, start first an interactive session on Snowy, for instance, as follows:

```
> interactive -M snowy -n 1 -t 0:10:00 -A uppmax2023-2-13
```

The above command will open a 10 minutes interactive session using one node.

Parallel performance analyser `Allinea MAP`

The performance analyser is a part of `Allinea` and its usage is highly recommended to better understand the performance and the scalability of your code.