MPI Software on the TeideHPC cluster

Introduction

This information is intended to help you compile and run MPI applications on the cluster cluster.

The Message Passing Interface (MPI) library allows processes in your parallel application to communicate with one another by sending and receiving messages. There is no default MPI library in your environment when you log in to the cluster. You need to choose the desired MPI implementation for your applications. This is done by loading an appropriate MPI module.

Currently the available MPI implementations on our cluster are OpenMPI, Mpich. For both implementations the MPI libraries are compiled and built with either the Intel compiler suite or the GNU compiler suite. These are organized in software modules.

MPI Intel compiler suite

To use Intel C/C++ and Fortran compilers, alongside Intel MPI, you only need to load the module *iimpi* (iimpi is a collection of tools.)

```
$ ml spider iimpi
----------------------------------------------------------------------------------------------------
  iimpi:
----------------------------------------------------------------------------------------------------
    Description:
      Intel C/C++ and Fortran compilers, alongside Intel MPI.

     Versions:
        iimpi/2021b
        iimpi/2022b

----------------------------------------------------------------------------------------------------
  For detailed information about a specific "iimpi" package (including how to load the
modules) use the module's full name.
  Note that names that have a trailing (E) are extensions provided by other modules.
  For example:

     $ module spider iimpi/2022b
----------------------------------------------------------------------------------------------------
```

```
$ module load iimpi/2022b
$ module list
```

```
Currently Loaded Modules:
  1) GCCcore/12.2.0   2) zlib/1.2.12   3) binutils/2.39   4) intel-compilers/2022.2.1   5)
numactl/2.0.16   6) UCX/1.13.1   7) impi/2021.7.1   8) iimpi/2022b
```

MPI GNU compiler suite

To use GNU Compiler Collection (GCC) including OpenMPI for MPI support you only need to load the module *gompi/2022b* (gompi is a collection of tools)

```
$ ml spider gompi/2022b

----------------------------------------------------------------------------------------------------------------------------------------
  gompi: gompi/2022b
----------------------------------------------------------------------------------------------------------------------------------------
    Description:
      GNU Compiler Collection (GCC) based compiler toolchain, including OpenMPI for MPI
support.


    This module can be loaded directly: module load gompi/2022b

    Help:
      Description
      ===========
      GNU Compiler Collection (GCC) based compiler toolchain,
       including OpenMPI for MPI support.


      More information
      ================
       - Homepage: (none)
```

```
$ module load gompi/2022b
$ module list

Currently Loaded Modules:
  1) GCCcore/12.2.0   3) binutils/2.39             5) numactl/2.0.16   7) impi/2021.7.1   9) GCC/
12.2.0  11) libxml2/2.10.3     13) hwloc/2.8.0       15) slurm/teide  17) OpenMPI/4.1.4
  2) zlib/1.2.12      4) intel-compilers/2022.2.1   6) UCX/1.13.1       8) iimpi/2022b    10) XZ/
5.2.7    12) libpciaccess/0.17  14) libfabric/1.16.1  16) UCC/1.1.0    18) gompi/2022b
```

Basic example for MPI applicatin.

This MPI job will start the parallel program *myapp.exe* with 12 processes in 2 nodes.

```
#!/bin/bash

### Job name
```

```
#SBATCH --job-name=MPIJOB
### File for the output
#SBATCH --output=MPIJOB_OUTPUT
### Time your job needs to execute, e. g. 50 min
#SBATCH --time=00:50:00
### Memory your job needs per node, e. g. 250 MB
#SBATCH --mem=250M

### Use more than one node for parallel jobs on distributed-memory systems, e. g. 2
#SBATCH --nodes=2

### Number of CPUS per task (for distributed-memory parallelisation, use 1)
#SBATCH --cpus-per-task=1
### Disable hyperthreading by setting the tasks per core to 1
#SBATCH --ntasks-per-core=1
### Number of processes per node, e. g. 6 (6 processes on 2 nodes = 12 processes in total)
#SBATCH --ntasks-per-node=6
# sandy, ilk (icelake)... arquitecture
#SBATCH --constrains=sandy
# ------------------------------------------------------------

### The last part consists of regular shell commands:
### Set the number of threads in your cluster environment to 1, as specified above
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

### Change to working directory
cd /home/usr/workingdirectory

module load gompi/2022b

### Run your parallel application
srun myapp.exe
```

It is very important to know the differences between *ntask, ntask-per-core,cpu-per-task ...*

Visit this page

You can check examples in our public repository on github https://github.com/hpciter