

Software MPI en el clúster TeideHPC

Introducción

Esta información está destinada a ayudarlo a compilar y ejecutar aplicaciones MPI en el clúster TeideHPC.

Las librerías de Interfaz de paso de Mensajes (MPI) permite que los procesos de su aplicación paralela se comuniquen entre sí mediante el envío y la recepción de mensajes. No hay una biblioteca MPI predeterminada en su entorno cuando inicia sesión en el clúster. Debe elegir la implementación MPI deseada para sus aplicaciones. Esto se hace cargando un módulo MPI apropiado.

Actualmente, las implementaciones de MPI disponibles en nuestro clúster son OpenMPI, Mpich. Para ambas implementaciones, las bibliotecas MPI se compilan y construyen con el conjunto de compiladores Intel o el conjunto de compiladores GNU y éstas están organizados en **módulos de software**.

Compiladores MPI intel

Para usar los compiladores de intel junto a Intel MPI, sólo necesitas cargar el modulo *iimpi* (iimpi carga una colección de módulos)

```
$ 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

Compiladore GNU para MPI

Para usar la colección de módulos con soporte OpenMPI bajo el compilador, sólo necesita cargar el módulo *gompi/2022b* (gompi es una colección de distintos módulos)

```
$ ml spider gompi
```

```
-----  
gompi: gompi/2021b  
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

Ejemplo básico para una aplicación MPI.

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)... architecture
#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
```

 **Es importante conocer las diferencias entre los parámetros de slurm *ntask*, *ntask-per-core*, *cpu-per-task* ...**

Visite [esta página](#) para entender su uso.

Puedes ver más ejemplos en nuestro repositorio público en github https://github.com/hpciter/user_codes