

# MPI

MPI (The Message Passing Interface), conçue en 1993-94, est une norme définissant une bibliothèque de fonctions, utilisable avec les langages C/C++ et Fortran. Elle permet d'exploiter des ordinateurs distants ou multiprocesseur par passage de messages.

## Principe

- Le programme est écrit dans un langage classique (Fortran, C, C++, etc.)
- chaque processus exécute éventuellement des parties différentes d'un programme
- toutes les variables du programme sont privées et résident dans la mémoire locale
- une donnée est échangée entre deux ou plusieurs processus via un appel à des fonctions particulières

## Exemples

## Environnement

Exemple C	Exemple fortran
<pre> Hello.c #include "mpi.h" #include &lt;stdio.h&gt;  int main(argc,argv) int argc; char *argv[]; { int numtasks, rank, rc;  rc = MPI_Init(&amp;argc,&amp;argv); if (rc != MPI_SUCCESS) { printf ("Error starting MPI program. Terminating.\n"); MPI_Abort(MPI_COMM_WORLD, rc); }  MPI_Comm_size(MPI_COMM_WORLD,&amp;numtasks); MPI_Comm_rank(MPI_COMM_WORLD,&amp;rank); printf ("Number of tasks= %d My rank= %d\n", numtasks,rank);  /***** do some work *****/  MPI_Finalize(); } </pre>	<pre> Hello.f program simple include 'mpif.h'  integer numtasks, rank, ierr, rc  call MPI_INIT(ierr) if (ierr .ne. MPI_SUCCESS) then print *,'Error starting MPI program. Terminating.' call MPI_ABORT(MPI_COMM_WORLD, rc, ierr) end if  call MPI_COMM_RANK(MPI_COMM_WORLD, rank, ierr) call MPI_COMM_SIZE(MPI_COMM_WORLD, numtasks, ierr) print *, 'Number of tasks=',numtasks,' My rank=',rank  C *****/ do some work *****/  call MPI_FINALIZE(ierr)  end </pre>

## Compilation

Il faut tout d'abord charger le module Open MPI (ompi)

```
$ module load mpi/openmpi/icc/1.7.5
```

### Pour le code C/C++

```
$ mpicc hello.c -o hello
```

### Pour le code Fortran

```
$ mpif90 hello.f -o hello
```



mpicc et mpif90 sont des wrappers autour du compilateur intel. i.e. le compilateur intel (ifort, icc, ...) sera utilisé pour

1 compiler le programme.

Afficher les options de compilation par défaut :

```
$ mpicc --showme  
icc -I/Softs/openmpi-intel-1.3.4/include -pthread -L/Softs/openmpi-  
intel-1.3.4/lib -lmpi -lopen-rte -lopen-pal -ldl -Wl,--export-dynamic -lnsl  
-lutil -shared-intel
```

```
$ mpif90 --showme  
ifort -I/Softs/openmpi-intel-1.3.4/include -I/Softs/openmpi-intel-1.3.4/lib  
-L/Softs/openmpi-intel-1.3.4/lib -lmpi_f90 -lmpi_f77 -lmpi -lopen-rte -  
lopen-pal -ldl -Wl,--export-dynamic -lnsl -lutil -shared-intel
```

## Exécution



Il est préférable de tester vos programmes MPI avant de les lancer avec SGE.

Exécution interactive sur la machine de login (mesocluster) :

```
$module load mpi/openmpi/icc/1.7.5  
$ mpirun -np 4 ./monAppli
```

Cela exécutera monAppli en utilisant 4 coeurs sur la machine mesocluster.

## Exécution avec SGE

Plus d'infos [Open MPI avec SGE](#) et [Utilisation SGE](#)

## Ressources

- [MPI Java Binding](#)
- Introduction à MPI [mpi-1.pdf](#)
- Message Passing Interface Tutorial <https://computing.llnl.gov/tutorials/mpi/>

From:

<http://mesowiki.univ-fcomte.fr/dokuwiki/> - **Wiki Utilisateurs - Mésocentre de calcul de Franche-Comté**

Permanent link:

<http://mesowiki.univ-fcomte.fr/dokuwiki/doku.php/mpi>

Last update: **2018/10/22 16:13**