Supercomputing

Hands-On 5: Basic Hybrid Application MPI+OpenMP

2022

Índice

1. Introduction

This Hands-on comprises 2 session. Next table shows the documents and files needed to develop in the exercises.

Session 1	Hello World	`hello-mpi+openmp.c`
Session 2	Hybrid Matrix Multiple	`mm-mpi.c`, `mm-openmp.c`, and `mm-openmp+mpi.c`

First, we will execute a sample hybrid code *Hello World* to understand the meaning of heterogeneous environment applied in Parallel Computing.

In order to compile and execute the program. At run time, an argument can be used to pass the number of threads for the algorithm. For example, to use the number of processes 4 you can use:

```
$ mpicc hello-mpi+omp.c -o hello-mpi+omp -fopenmp
```

```
$ mpirun -np 4 ./hello-mpi+omp
```

The result of the execution will be shown on the screen.

```
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>

int main( int argc, char *argv[])
{
  int nthreads,nprocs,idpro,idthr;
  int  namelen;
  char processor_name[MPI_MAX_PROCESSOR_NAME];

  MPI_Init(&argc,&argv);
  MPI_Comm_size(MPI_COMM_WORLD,&nprocs);
  MPI_Comm_rank(MPI_COMM_WORLD,&idpro);
  MPI_Get_processor_name(processor_name,&namelen);

#pragma omp parallel private(idthr) firstprivate(idpro,processor_name)
{
 idthr = omp_get_thread_num();
 printf("Hello World  thread %d, From %d processor %s\n",idthr,idpro,processor_name);
 if (idthr == 0)
 {
  nthreads = omp_get_num_threads();
  printf("Master processor %d, number of threads %d, number of processor %d\n",idpro,
                                                                          nthreads,
                                                                           nprocs);
 }
}
  MPI_Finalize();
}
```

Figura 1: Hybrid code for the sample Hello World.

2. Matrix Multiple Benchmarks

2.1. MPI

In order to compile MPI program, we should include the proper compilation option, such as:

```
$ mpicc mm-mpi.c -o mm-mpi
```

To execute an MPI program with several processes, you can input by command line:

```
$ mpirun -np 4 ./mm-mpi 100
```

2.2. OpenMP

In order to compile OpenMP program, we should include the proper compilation option, such as:

```
$ gcc mm-openmp.c -o mm-openmp -fopenmp
```

To execute an OpenMP with several threads, you can use the following example (changing the number of threads accordingly):

```
$ OMP_NUM_THREADS=16 ./mm-openmp  100  1000 100
```

2.3. MPI+OpenMP

Please remind that in order to compile OpenMP with MPI programs, we should include the proper compilation option, such as:

```
$ mpicc mm-openmp+mpi.c -o mm-openmp+mpi -fopenmp
```

To execute an OpenMP with MPI program with several threads and processes, you can input by command line:

```
$ mpirun -np 4 ./mm-openmp+mpi 1000 16
```

3. Practice with the Benchmarks

From the parallelized matrix multiplication codes (MPI (`mm-mpi.c`), OpenMP (`mm-openmp.c`) and Hybrid (`mm-openmp+mpi.c`)). Make the following plots:

- Problem size Execution time
- Speedup

They are comparatively commenting on the performance of the optimizations of each code, with each group doing the experimentation on a specific execution platform.

Referencias

[1] M. Boratto. *Hands-On Supercomputing with Parallel Computing*. Available: `https://github.com/muriloboratto/Hands-On-Supercomputing-with-Parallel-Computing`. 2022.

[2] Forum, Message Passing Interface. *MPI: A Message-Passing Interface Standard*. University of Tennessee, 1994, USA.

[3] B. Chapman, G. Jost and R. Pas. *Using OpenMP: Portable Shared Memory Parallel Programming*. The MIT Press, 2007, USA.