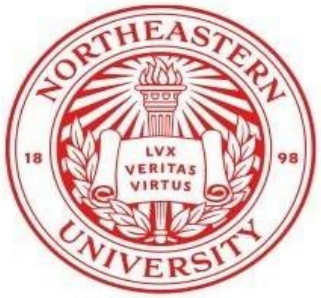


Northeastern University

CSYE 7374 - PARALLEL MACHINE LEARNING AND AI

SUMMER 2020

HOMEWORK 1

SRISHTI ASHOK MISHRA

001305178

Part 1:

Login into discovery: ssh -Y mishra.sr@login.discovery.neu.edu

Using mkdir command create a folder : csye7374-mishra.sr and subfolder : homework1

Compute node : srun –pty /bin/bash

Compiled OMP programs by

module load gcc

gcc -o omp_hello -fopenmp omp_Hello.c

export OMP_NUM_THREADS=8

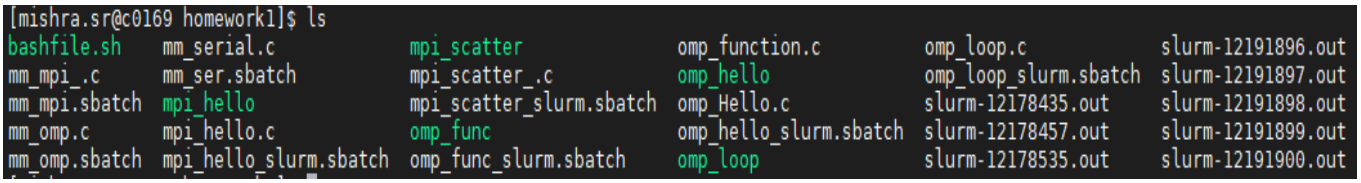
./omp_hello

Compiled MPI Programs by

module load openmpi

mpicc -o mpi_scatter mpi_scatter_.c

mpirun -np 4 --oversubscribe mpi_scatter



Part 2 : Slurm Script

Omp_Hello

```
#!/bin/bash

##Normal configurations

##slurm script
#SBATCH --job-name=omp_hello //name
#SBATCH --output=omp_hello.output
#SBATCH --error=omp_hello.err

#SBATCH --partition express
#SBATCH --nodes=1
#SBATCH --time=0:05:00

work=$HOME/csye7374-mishra.sr/homework1
cd $work
module load gcc
gcc -o omp_hello -fopenmp omp_Hello.c
export OMP_NUM_THREADS=8
./omp_hello
```

Omp_func

```
#!/bin/bash

##Normal configurations

##slurm script
#SBATCH --job-name=omp_func //name
#SBATCH --output=omp_func.output
#SBATCH --error=omp_func.err

#SBATCH --nodes=1
#SBATCH --partition express
#SBATCH --time=05:00

work=$HOME/csye7374-mishra.sr/homework1
cd $work
module load gcc

gcc -o omp_func -fopenmp omp_function.c
export OMP_NUM_THREADS=8
./omp_func
```

Omp_loop

```
#!/bin/bash

##Normal configurations

##slurm script
#SBATCH --job-name=omp_loop //name
#SBATCH --output=omp_loop.output
#SBATCH --error=omp_loop.err
#SBATCH --nodes=1
#SBATCH --partition express

#SBATCH --time=05:00

work=$HOME/csye7374-mishra.sr/homework1
cd $work
module load gcc

gcc -o omp_loop -fopenmp omp_loop.c
export OMP_NUM_THREADS=8
./omp_loop
```

Mpi_hello

```
#!/bin/bash

##Normal configurations

]##slurm script
#SBATCH --job-name=mpi_hello //name
#SBATCH --output=mpi_hello.output
#SBATCH --error=mpi_hello.err
#SBATCH --nodes=1
#SBATCH --partition express

#SBATCH --time=05:00

work=$HOME/csye7374-mishra.sr/homework1
cd $work

module load openmpi
mpicc -o mpi_hello mpi_hello.c
mpirun --mca btl_base_warn_component_unused 0 -np 4 --oversubscribe mpi_hello
```

Mpi_scatter

```
#!/bin/bash

##Normal configurations

##slurm script
#SBATCH --job-name=mpi_scatter //name
#SBATCH --output=mpi_scatter.output
#SBATCH --error=mpi_scatter.err
#SBATCH --partition express
#SBATCH --nodes=1
#SBATCH --time=0:05:00

work=$HOME/csye7374-mishra.sr/homework1
cd $work

module load openmpi
mpicc -o mpi_scatter mpi_scatter_.c
mpirun --mca btl_base_warn_component_unused 0 -np 4 --oversubscribe mpi_scatter
```

Bash file

```
#!/bin/bash

##sbatch of all the 5 files

sbatch mpi_hello_slurm.sbatch
sbatch mpi_scatter_slurm.sbatch
sbatch omp_hello_slurm.sbatch
sbatch omp_loop_slurm.sbatch
sbatch omp_func_slurm.sbatch
```

Part 3 and 4

mm_mpi.c

Elapsed time and matrix changes

```c
/******************************************************************************
* FILE: mm_mpi.c
* DESCRIPTION:
*   Matrix Multiply - MPI implementation
*   In this code, the master task distributes a matrix multiply
*   operation to numtasks-1 worker tasks.
******************************************************************************/
#include "mpi.h"
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define NRA 500                 /* number of rows in matrix A */
#define NCA 500                 /* number of columns in matrix A */
#define NCB 300                  /* number of columns in matrix B */
#define MASTER 0                /* taskid of first task */
#define FROM_MASTER 1           /* setting a message type */
#define FROM_WORKER 2           /* setting a message type */

int main (int argc, char *argv[])
{
int numtasks,              /* number of tasks in partition */
    taskid,                /* a task identifier */
    numworkers,            /* number of worker tasks */
    source,                /* task id of message source */
    dest,                  /* task id of message destination */
    mtype,                 /* message type */
    rows,                  /* rows of matrix A sent to each worker */
    averow, extra, offset, /* used to determine rows sent to each worker */
    i, j, k, rc;           /* misc */
double  a[NRA][NCA],           /* matrix A to be multiplied */
    b[NCA][NCB],           /* matrix B to be multiplied */
    c[NRA][NCB];           /* result matrix C */

/*double starttime, endtime, totaltime;*/
double time_spent = 0.0;
clock_t begin = clock();
```

```c
                MPI_COMM_WORLD, &status);
            printf("Received results from task %d\n",source);
        }


clock_t end = clock();
time_spent += (double)(end - begin) / CLOCKS_PER_SEC;
time_spent = time_spent *1000;
    printf("Time elpased is %f milliseconds\n", time_spent);


 /* Print results */
      printf("******************************************************\n");
      printf("Result Matrix:\n");
      for (i=0; i<NRA; i++)
      {
         printf("\n");
         for (j=0; j<NCB; j++)
            printf("%6.2f   ", c[i][j]);
      }
      printf("\n******************************************************\n");
      printf ("Done.\n");
   }
```

mm_mpi.sbatch

```
#!/bin/bash

##Normal configurations

##slurm script
#SBATCH --job-name=mpi //name
#SBATCH --output=mpi.output
#SBATCH --error=mpi.err
#SBATCH --time=0-00:05:00 //time for 5 mins

## Parallel Configurations
#SBATCH --partition express
#SBATCH -N 1



work=$HOME/csye7374-mishra.sr/homework1
cd $work

module load openmpi

mpicc -o mm_mpi mm_mpi_.c

echo " With Core 2"
echo "-------------------------------------------------------------------------------"
mpirun --mca btl_base_warn_component_unused 0 -np 2 --oversubscribe mm_mpi
echo " With Core 4"
echo "-------------------------------------------------------------------------------"
mpirun --mca btl_base_warn_component_unused 0 -np 4 --oversubscribe mm_mpi
echo "With Core 8"
echo "-------------------------------------------------------------------------------"
mpirun --mca btl_base_warn_component_unused 0 -np 8 --oversubscribe mm_mpi
```

mm_omp.c

```
/******************************************************************************
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define NRA 500                 /* number of rows in matrix A */
#define NCA 500                 /* number of columns in matrix A */
#define NCB 300                 /* number of columns in matrix B */

int main (int argc, char *argv[])
{
int tid, nthreads, i, j, k, chunk;

double  a[NRA][NCA],           /* matrix A to be multiplied */
    b[NCA][NCB],           /* matrix B to be multiplied */
    c[NRA][NCB];           /* result matrix C */

chunk = 10;                    /* set loop iteration chunk size */
double time_spent = 0.0;
clock_t begin = clock();
```

```
clock_t end = clock();
time_spent += (double)(end - begin) / CLOCKS_PER_SEC;
time_spent = time_spent*1000;

    printf("Time elpased is %f milliseconds\n", time_spent);


/*** Print results ***/
printf("******************************************************\n");
printf("Result Matrix:\n");
for (i=0; i<NRA; i++)
  {
  for (j=0; j<NCB; j++)
    printf("%6.2f   ", c[i][j]);
  printf("\n");
  }
printf("******************************************************\n");
printf ("Done.\n");

}
```

mm_omp.sbatch

```
#!/bin/bash

##Normal configurations

##slurm script
#SBATCH --job-name=omp //name
#SBATCH --output = omp.output
#SBATCH --error = omp.err
#SBATCH --partition=express
#
# Time format = HH:MM:SS, DD-HH:MM:SS
#SBATCH --time=05:00
#SBATCH -N 1

work=$HOME/csye7374-mishra.sr/homework1
cd $work

module load gcc

gcc -o mm_omp -fopenmp mm_omp.c

export OMP_NUM_THREADS=2
./mm_omp
export OMP_NUM_THREADS=4
./mm_omp
export OMP_NUM_THREADS=8
./mm_omp
```

mm_serial.c

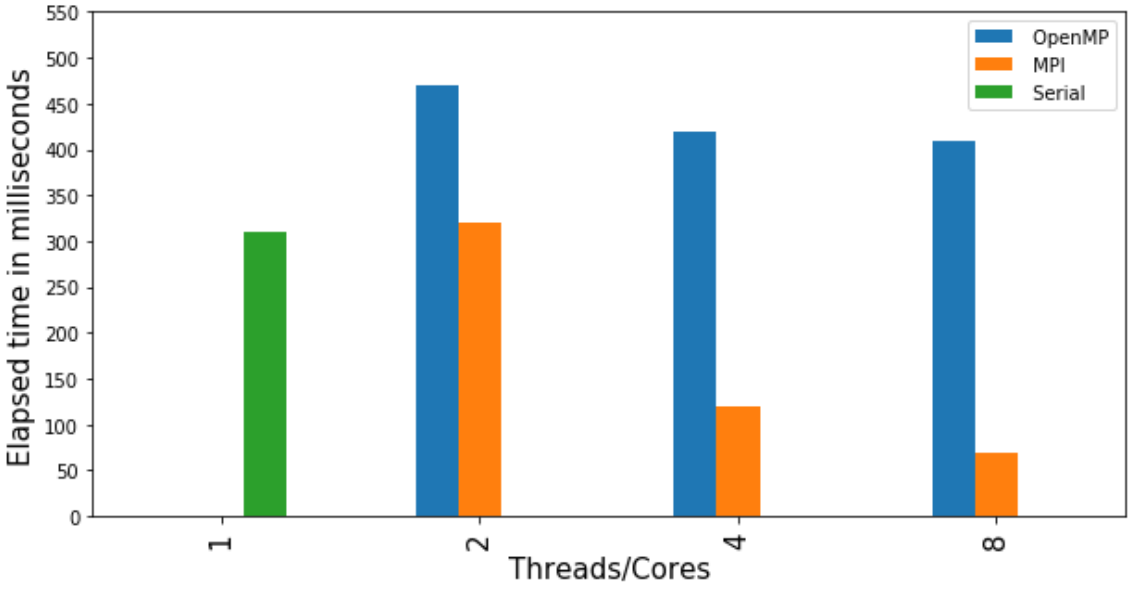
```c
*   Serial Matrix Multiply - C Version
******************************************************************
#include <stdio.h>
#include <stdlib.h>
#include <time.h>


#define NRA 500         /* number of rows in matrix A */
#define NCA 500         /* number of columns in matrix A */
#define NCB 300         /* number of columns in matrix B */

int main(int argc, char *argv[])
{
int     i, j, k, chunk;           /* misc */
double a[NRA][NCA],          /* matrix A to be multiplied */
       b[NCA][NCB],          /* matrix B to be multiplied */
       c[NRA][NCB];      /* result matrix C */

       chunk = 10;
       double time_spent = 0.0;
clock_t begin = clock();
```

mm_ser.sbatch

```
#!/bin/bash

##Normal configurations

##slurm script
#SBATCH --job-name=mpi //name
#SBATCH --output=mpi.output
#SBATCH --error=mpi.err

## Parallel Configurations

#SBATCH --nodes 1
#SBATCH --partition express
#SBATCH --time=05:00

work=$HOME/csye7374-mishra.sr/homework1
cd $work
module load gcc

gcc -o mm_ser -fopenmp mm_serial.c
./mm_ser
```

Tabulated results:

Cores/threads	OpenMP	MPI	Serial	
1			310	
2	470	320		
4	420	120		
8	410	70		


OpenMP
MPI
Serial
Elapsed time in milliseconds
550
500
450
400
350
300
250
200
150
100
50
0
1
2
4
8
Threads/Cores