Honesty Guaranty

I know the examination rules, promise to be honest and abide by the rules. Signature: _____

Examination of Northwestern Polytechnical University (A)

2022 — 2023 School Year 1 Semester

School of Computer Science, Course: Parallel Programming , Class Hours: 48
Exam. Date: 2022.11.28 , Exam. Duration: 2 Hours, Written Exam. (Open-book)

Item	I	II	III	IV	V	Total Score
Score						

Class		Student ID.		Name	

I. Provide a very brief definition of the following terms (20 points, 4 points per item)

1. Flynn's taxonomy of computers

2. Cache coherence

3. thread safety

4. One sided communication

5. PGAS

II. Circle TRUE or FALSE based on the statements. (12 points, 2 points per item)

1. Currently, the main architecture of parallel computers is cluster …TRUE/FALSE
2. MPI programs can only be SPMD …TRUE/FALSE
3. POSIX threads (Pthreads) are using compiler directives …TRUE/FALSE
4. Hybrid model of MPI and CUDA can be used for GPU clusters …TRUE/FALSE
5. Communication overhead may be a major performance challenge … TRUE/FALSE
6. Amdahl's law is optimistic for the scalability of parallel programs ... TRUE/FALSE

III. Choose the right answer. (24 points, 4 points per item)

1. Suppose the parallel fraction of a program is 0.8, according to Gustafson's law, what is the speedup if we use 100 processors to run this parallel program? ( )

   A. 1.25 B. 5 C. 80.2 D. 100

2. Which of the following about OpenMP is incorrect? ( )

   A. OpenMP is an API that enables explicit multi-threaded parallelism.

   B. The primary components of OpenMP are compiler directives, runtime library, and environment variables.

   C. OpenMP is designed for distributed memory parallel systems and guarantees efficient use of memory.

   D. OpenMP supports UMA and NUMA architectures.

3. Which operation ( ) does not belong to collective communication?

   A. MPI_Scatter
   B. MPI_Wait
   C. MPI_Barrier
   D. MPI_Reduce

4. Which operation ( ) belongs to non-blocking communication of MPI?

   A. MPI_Ssend
   B. MPI_Bsend
   C. MPI_Rsend
   D. MPI_Isend

5. What is the final output about *n* after running? ( )

```
int n=10, i;
omp_set_num_threads(2);
#pragma omp parallel for firstprivate(n)
for (i=1; i<4; i++)
   n=n+2*i;
printf("n= %d\n", n);
```

A. n=10 B. n=11 C. n=19 D. n=26

6. We need to use each thread to calculate one output element of a vector addition. Assume that variable i should be the index for the element to be processed by a thread. What would be the expression for mapping the thread/block indices to data index?
   A. i=threadIdx.x + threadIdx.y;
   B. i=blockIdx.x + threadIdx.x;
   C. i=blockIdx.x*blockDim.x + threadIdx.x;
   D. i=blockIdx.x * threadIdx.x;

IV. Answer the questions briefly (15 points, 5 points per item)

1. What are the main parallel overheads?

2. What is the False sharing? And how do you solve this problem?

3. What are the advantages and limitations of hybrid MPI and OpenMP programming model?

V. Programming (29 points)

1. The following C++ program calculates the value of Pi. Please complete the missing lines of code. (**9 points**)

```
#include <iostream>
#include "mpi.h"
#include <ctime>
#include <cmath>
using namespace std;

const int N = 1000000;
double start, finish;
int main (int argc, char *argv[]) {
int numprocs, myid;
/* Please complete the missing lines of code using MPI Routines here.*/
```

_____

_____

_____

```
   start = MPI_Wtime();
   double partSum=0.0;
   double Pi = 0.0;
for (int i = myid; i < N; i += numprocs) {
      partSum += sqrt (1 – (double(i) / N) * (double (i) / N)) / N;
      }
/* Please complete the missing line of code using a MPI Routine here.*/
```

_____

```
cout << "Myid" << myid << ", partSum:" << partSum << endl;
if (myid == 0)
{
Pi *= 4.0;
finish = MPI_Wtime();
cout << "The value of Pi is : " << Pi << endl;
cout << "#" << numprocs << "run time : " << finish – start << endl;
}
/* Please complete the missing line of code using a MPI Routine here.*/
```

_____

```
return 0;
}
```

2. (**6 points**) Consider the problem of counting the array a. the final summation of a is stored in variable sum. Please read the following code and answer questions.

```
#include <stdio.h>
#include <omp.h>
#define N 10
int main(void) {
  int i, sum=0, a[N];
  for(i=0;i<N;i++)
     a[i] = i;
  #pragma omp parallel for shared(a,sum) private(i)
  for(i=0;i<N;i++)
      sum += a[i];
  printf("sum = %d\n",sum);
  return 0;
}
```

Questions:

1) Do you think the program can get the correct result? If not, can you explain it? (**3 points**)

2) Propose a solution to overcome the problem in the above example. (**3 points**)

3. Consider the following OpenMP program and answer questions. (8 points)

```
#include <stdio.h>
#include <omp.h>
#define COLUMNS 3
#define ROWS 3
int m[ROWS][COLUMNS];
void fillColumn(int j){
    int i;
    #pragma omp for
    for (i = 0; i < ROWS; i++)
        m[i][j] = omp_get_thread_num();
}

int main(){
    int i, j;
    for (i = 0; i < ROWS; i++) // initialize the array
        for (j= 0; j < COLUMNS; j++)
            m[i][j] = -1;

    #pragma omp parallel num_threads(COLUMNS-1)
        fillColumn(0);

    #pragma omp parallel for num_threads(COLUMNS-1)
    for (j = 1; j < COLUMNS; j++)
        fillColumn(j);

    for (i = 0; i < ROWS; i++) // print out the results
    {
        for (j= 0; j < COLUMNS; j++)
            printf(" %2d ", m[i][j]);
        printf("\n");
    }
    return 0;
}
```

Questions:

1) Supposing that you use gcc compiler, what the compiler command is to get an executable openmp program? (**2 points**)

2) What is the output of this program using openmp? (**6 points**)

4. Consider the following program. Explain how you would parallelize the following segment of C code. Modify the code and insert the necessary OpenMP pragma(s) into your code. (**6 points**)

```
#define X 50
#define Y 50
int i, j;
double a[X,Y];
double m;
for (i=1; i< X; i++)
      for (j= 0; j<Y; j++) {
            m = i*j;
            a[i,j]=tmp*a[i-1,j];
}
```