|  |
| --- |
| **Honesty Guaranty**  **I know the examination rules, promise to be honest and abide by the rules. Signature：** |

**Examination of Northwestern Polytechnical University (A)**

**2021 － 2022 School Year 1 Semester**

|  |
| --- |
|  |

**School of Computer Science, Course: Parallel Programming , Class Hours: 48**

**Exam. Date: 2021.12.26 , 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. Gustafson’s Law
2. CCNUMA
3. BSP
4. CUDA
5. Cartesian topology

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

1. Functional decomposition is a form of task parallelism ... TRUE/FALSE
2. MPI programs cannot run on shared-memory systems …TRUE/FALSE
3. POSIX threads (Pthreads) programming model is library based …TRUE/FALSE
4. The current No. 1 Supercomputer in the world is IBM Summit …TRUE/FALSE
5. Too many barriers may be a major performance challenge … TRUE/FALSE
6. Performance analysis tool TAU can analyze binary codes ... TRUE/FALSE

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

1. Suppose the parallel fraction of a serial program is only 0.2, according to Amdahl’s law, what is the limit of speedup? ( )
2. 2.5 B. 4 C. 5 D. 1.25
3. In order to remove the unnecessary synchronization barrier, what statement should we use correctly? ( )
4. #pragma omp for
5. #pragma omp single
6. #pragma omp sections
7. #pragma omp for nowait
8. Which operation ( ) does not belong to point-to-point communication?

A. MPI\_Send

B. MPI\_Bsend

C. MPI\_Barrier

D. MPI\_Irecv

1. Which type of operation ( ) does not belong to collective opeartions of MPI?

A. data movement

B. synchornization

C. collective computation

D. communicator management

1. What are the final results after running? ( )

int x=5, i;

omp\_set\_num\_threads(5);

#pragma omp parallel for firstprivate(x)

for (i=1; i<5; i++)

x=x+2\*i;

A. x=5 B. x=7 C. x=11 D. x=13

1. There are 12 iterations (index from 0 to 11) of for loop in OpenMP program. Suppose we use 3 threads to share the loop construct, and always assign the iteration index with the following pattern: Thread 0: 0, 1, 6, 7; Thread 1: 2, 3, 8, 9; Thread 2: 4, 5, 10, 11.

What schedule clause should use? ( )

A. schedule(static)

B. schedule(static,2)

C. schedule(dynamic)

D. schedule(dynamic,2)

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

1. What are the main considerations for partitioning design when designing parallel algorithms?
2. Please briefly compare OpenMP and Pthreads by describing the advantages and disadvantages of each theads model.
3. What are the differences between shared memory and message passing programming paradigms? Please describe the pros and cons of them.

**V. Programming (29 points)**

1. The following program uses Non-blocking MPI routines. Please complete the missing lines of code. ( **9 points**)

#include "mpi.h"

#include <stdio.h>

int main (int argc, char \*argv[])

{

int numtasks, rank, next, prev, buf[2], tag1=1, tag2=2;

MPI\_Request reqs[4];

MPI\_Status stats[4];

**/\* Please complete the missing lines of code using MPI Routines here.\*/**

Prev = rank-1;

next = rank+1;

if (rank == 0) prev= numtask-1;

if (rank == (numtask -1)) prev= 0;

MPI\_Irecv (&buf[0], 1, MPI\_INT, prev, tag1, MPI\_COMM\_WORLD, &regs[0]);

**/\* Please complete the missing line of code using MPI Routine here.\*/**

MPI\_Isend (&rank, 1, MPI\_INT, prev, tag2, MPI\_COMM\_WORLD, &regs[2]);

**/\* Please complete the missing line of code using MPI Routine here.\*/**

printf(“ Process %d of %d\n”, rank, numtasks);

**/\* Please complete the missing line of code using MPI Routine here.\*/**

MPI\_Finalize();

}

1. (**6 points**) Consider the problem of counting the number of 5 in ***a[]***using *t* number of threads. Suppose that length can be evenly divided by *t*. Function *counting* is the thread function, *length* is the size of *a[]*.

The following solution has been proposed:

**void\*** **counting(void\* thread\_id){**

**int id = (int)thread\_id**

**int length\_per\_thread = length/t;**

**int start = id \* length\_per\_thread;**

**int i;**

**for (i= start; i< start+length\_per\_thread; i++)**

**{**

**if (a[i] == 5)**

**count +=1;**

**}**

**}**

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. Read the program *main.c* and answer the questions. (**8 points**)

#include <stdio.h>

#include <stdlib.h>

#include <omp.h>

void main() {

int x = 0;

int y = 0;

int i;

omp\_set\_num\_threads(4);

#pragma omp parallel

{

#pragma omp atomic

x++;

#pragma omp master

{

#pragma omp critical

{

y++;

**printf("master: %d\n", omp\_get\_thread\_num( )); //Line-1**

}

}

#pragma omp single

{

#pragma omp critical

{

**printf("single: %d\n", omp\_get\_thread\_num( )); //Line-2**

y++;

}

}

}

**printf("x: %d, y: %d\n", x, y); //Line-3**

}

Questions:

1. If you use gcc compiler, what is the compiler command to get an executable openmp program? (**2 points**)
2. What are the outputs of Line-1, Line-2 and Line-3? (**6 points**)
3. 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 tmp;

for (i=1; i< X; i++)

for (j= 0; j<Y; j++) {

tmp = i\*j;

a[i,j]=tmp\*a[i-1,j];

}