WARWICK
THE UNIVERSITY OF WARWICK

UNIVERSITY OF WARWICK

Paper Details

Paper Code: CS9220_B

Paper Title: High Performance Computing

Exam Period: Summer 2022

Exam Rubric

Time Allowed: 2 hours

Exam Type: Standard Examination

Approved Calculators: Permitted

Additional Stationery

Instructions

Choose to answer TWO questions from Questions 1, 2 and 3.

Question 4 is **compulsory**.

Read carefully the instructions on the answer book and make sure that the particulars required are entered on each answer book.

WARWICK
THE UNIVERSITY OF WARWICK

1. *This question is about the fundamental knowledge.*

(a) Give an example of instruction-level parallelism and discuss the factors that limit the further improvement of the instruction-level parallelism. [7]

(b) What dependencies exist in the following sequence of statements? Further, how can the anti-dependency and the output dependency in these statements be removed? [10]

```
if(a≥0) {
        a=b+c;
        e=a+b;
}
else {
        c=a×b;
        d=a+e;
        b=a+d;
        d=a×c;
}
```

(c) Why can superthreading and hyperthreading improve the performance of the pipeline mechanism? [8]

(d) Top500 and Graph500 are two supercomputer lists in the world. What are the differences between these two lists? The answers should focus on the type of performance that the two lists target, benchmarking applications and their features, and the key architecture factors in the computer systems that have impact on the performance. [10]

2. *This question is about parallel programming models*

(a) How can synchronised statements provide finer-grained synchronisation than synchronised methods in Java? [8]

(b) How are the data to be sent expressed in an MPI communication operation? Also, why are *message tag* and *communicator* the necessary parameters in MPI_Send? [7]

(c) A collective communication operation is performed by all relevant processes at the same time with the same set of parameters. However, different processes may interpret the parameters differently. Describe, using illustrative examples if necessary, the operations of the following two MPI collective communication calls. Further, discuss how different processes interpret different parameters in these functions.

i) MPI_Bcast (*void *buffer, int count, MPI_Datatype datatype, int root, MPI_Comm comm*) [6]

ii) MPI_Scatter(void **sendbuf*, int *sendcnt*, MPI_Datatype *sendtype*, void **recvbuf*, int *recvcount*, MPI_Datatype *recvtype*, int *root*, MPI_Comm *comm*) [6]

(d) In MPI, programmers are allowed to construct their own data types. MPI_Type_struct is one of the MPI functions serving this purpose. The format of the function is as follows:

MPI_Type_struct (int *count*,
int **array_of_blocklengths*,
int **array_of_displacements*,
MPI_Datatype **array_of_types*,
MPI_Datatype **newtype*)

Let *oldtype1*={(MPI_DOUBLE, 0), (MPI_CHAR, 8)} with the extent of 9 bytes and *oldtype2* ={(MPI_FLOAT, 0), (MPI_INT, 4)} with the extent of 6 bytes. Let *a*[2]={2, 3} and *b*[2]={4, 32}, c[2]={*oldtype1*, *oldtype2*}. Note that the unit of the elements in array *b* is *byte*.

What is the memory layout of *newtype* after calling the following function? MPI_Type_struct (2, *a*, *b*, *c*, *newtype*) [8]

3. *This question is about high performance computing systems.*

(a) What are the differences between Clusters and Massively Parallel Processing (MPP) systems? [8]

(b) What are the differences in the designing objectives between a GPU and a CPU? [10]

(c) What are the data sieving technique and the collective I/O? Also, how can the data sieving technique and collective I/O help improve the I/O performance? [10]

(d) The topology of node interconnection plays an important role in the performance of a HPC system. What are the definitions of *node degree* and *bisection width* in a network topology? What are the node degree and bisection width of a **3D** hypercube? Draw an exemplar **4D** hypercube. [7]

4. *This question is about performance analysis methods.* (**Note this question is compulsory**)

Assume that when a program performs computations at each point in a 3-Dimensional grid, it needs to obtain the data from neighboring points. The size of each dimension of the grid is *N*. The communication pattern among the grid points is shown in Figure 1.

i) Construct three performance models for the execution time of the program when the grid is partitioned using 1-D, 2-D and 3-D decompositions. Note that the communication intensity in the *z* direction of the grid is different from that in the *x* and *y* direction. You should apply sensible partition strategies when constructing the performance models. [15]

ii) Based on the constructed performance models, derive parallel efficiency and iso-efficiency functions under these three decomposition methods. [9]

iii) Which decomposition method is better in terms of scalability? Explain your answer from the perspective of their iso-efficiency functions. [6]

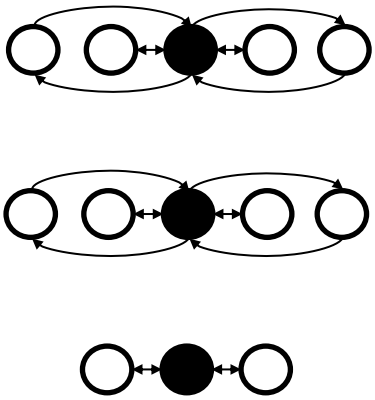

Figure 1. The communication pattern among the grid points in Question 4; a) the communication pattern among the points in the *x* direction; b) the communication pattern among the points in the *y* direction; c) the communication pattern among the points in the *z* direction