PRINCE OF SONGKLA UNIVERSITY
FACULTY OF ENGINEERING

Midterm Examination: Semester 1 **Academic Year**: 2014

Date: 18 October 2014 **Time**: 13.30 - 16.30 (3 hours)

Subject Number: 242-530 **Room**: R200

Subject Title: Parallel and Distributed Computing

Exam Duration: 3 hours

This paper has 15 pages, 10 questions and 180 marks (30%).

Authorised Materials:

- Writing instruments (e.g. pens, pencils).
- Textbooks, a notebook, handouts, and dictionaries are permitted.

Instructions to Students:

- Scan all the questions before answering so that you can manage your time better.
- Answers **must** be written in **Thai**.
- Write your name and ID on every page.
- Any unreadable parts will be considered wrong.

When drawing diagrams or coding, use good layout, and short comments; marks will not be deducted for minor syntax errors.

Cheating in this examination

Lowest punishment: Failed in this subject and courses dropped for next semester.

Highest punishment: Expelled.

NO	Time (Min)	Marks	Collected	NO	Time (Min)	Marks	Collected
1	25	25		6	12	12	
2	10	10		7	8	8	
3	20	20		8	15	15	
4	40	40		9	15	15	
5	20	20		10	15	15	
Total	180	180			30%		

Question 1 (25 marks; 25 minutes)

Tell whether the following statements are true (T) or false (F).

a) _____ Manual Parallelization is a time consuming, complex, error-prone, iterative process.

b) _____ In Automatic Parallelization, the system's performance, flexibility and results are better than Manual Parallelization.

c) _____ Automatic Parallelization may actually not parallelize code if the analysis suggests there are inhibitors or the code is too complex.

d) _____ "Embarrassingly parallel" means that problems can be decomposed and executed in parallel with virtually no need for tasks to share data so that very little inter-task communication is required.

e) _____ The problem that concerns independent calculations is non-parallelizable.

f) _____ Barrier is often implemented by establishing a synchronization point within an application where a task may not proceed further until another task(s) reaches the same or logically equivalent point.

g) _____ Synchronization usually involves waiting by at least one task, and can therefore cause a parallel application's wall clock execution time to increase.

h) _____ Uniform Memory Access (UMA) is represented by Symmetric Multiprocessor (SMP) machines which have identical processors

i) _____ Non-Uniform Memory Access (NUMA) means equal access and access times to memory.

j) _____ NUMA means one SMP can directly access memory of another SMP.

k) _____ Cache coherent means if one processor updates a location in shared memory, all the other processors do not know about the update.

l) _____ In Distributed Memory Architecture, there is no concept of global address space across all processors.

m) _____ In Distributed Memory Architecture, changes made to its local memory have no effect on the memory of other processors.

n) _____ In Shared Memory Architecture, processors have their own local memory.

o) _____ Changes in a memory location effected by one processor are visible to all other processors.

p) _____ The 2D Mesh Switched Network Topology s considered fully connected as there is a direct link between all pairs of nodes and also many simultaneous transfers are possible.

q) _____ Multi-stage Switched Network Topology provides a large number of parallel paths between sources and destinations.

r) _____ In Hypercube Switched Network Topology, node i connected to k nodes whose addresses differ from i in exactly one bit position.

s) _____ Crossbar Switched Network Topology allows communication only between neighboring switches.

t) _____ In Tree Switched Network Topology, the root node can become a

bottleneck and affects the performance unless applying the concept of hyper-tree.

u) _____ When spotting the bottlenecks and inhibitors in a program, investigate other algorithms if possible or restructure the program to reduce or eliminate unnecessary slow areas.

v) _____ Asynchronous communications do not require "handshaking" between tasks that are sharing data and are often referred to as non-blocking communications.

w) _____ Synchronous communications allow tasks to transfer data independently from one another and other work can be done while the communications are taking place.

x) _____ Barrier mechanism guarantees that no process will proceed beyond a designated point in the program, until every process has reached that point.

y) _____ In parallel computing, granularity is a qualitative measure of the ratio of communication to computation.

Question 2 (10 marks; 10 minutes)

Explain the following problems associated with shared data and how to solve the problems.

a) Cache Coherence (5 marks)

b) Race Condition (5 marks)

Question 3 (20 marks; 20 minutes)

Apply Amdahl's Law to answer the following questions.

a) Explain and demonstrate the effects and limitations of the parameters in the Amdahl's law (number of processors, problem size and proportion of parallel fraction). (12 marks)

b) Find the speedup when there are 2 processors and 40% parallelizable code (2 marks)

c) Find the speedup when there are 2 processors and 80% parallelizable code (2 marks)

d) Find the speedup when there are 16 processors and 40% parallelizable code (2 marks)

e) Find the speedup when there are 16 processors and 80% parallelizable code (2 marks)

Question 4 (40 marks; 40 minutes)

a) List 4 factors that contribute to parallel overhead. (4 marks)

b) List 4 hardware factors that play a significant role in scalability. (4 marks)

c) What and why do we need to identify when we are in the process of understanding the problem and the program which is the first step in developing parallel software? (6 marks)

d) List at least 4 factors to consider when designing your program's inter-task communications. (4 marks)

e) Compare *point-to-point* and *collective* communications and give examples of MPI commands used in each type. (4 marks)

Point-to-point Communication	Collective Communication

f) Compare *fine-grain* and *coarse-grain* Parallelism. (4 marks)

Fine-grain Parallelism	Coarse-grain Parallelism

g) Compare *loop independent data dependence* and *loop carried data dependence* by giving an example of code fragment for each type. (4 marks)

Loop independent data dependence	Loop carried data dependence

h) What is *load balancing*? Why is it important? (4 marks)

i) Give an example of problems that result in load imbalances even if data is evenly distributed among tasks. (2 marks)

Question 5 (20 marks; 20 minutes)

Answer the following questions.

a) Compare Parallel Vector Processor (PVP) and Symmetric Multiprocessors (SMP). (5 marks)

Parallel Vector Processor	Symmetric Multiprocessors

b) Compare the advantages and disadvantages of Shared Memory Model and Distributed Memory Model. (5 marks)

Shared Memory Model	Distributed Memory Model

c) Compare OpenMP and MPI. (5 marks)

OpenMP	MPI

d) List the significant constraints in building faster serial computers. (5 marks)

Question 6 (12 marks; 12 minutes)

Compare the following ways to program parallel computers.

a) Extend compilers (3 marks)

b) Extend languages (3 marks)

c) Add parallel language layer on top of sequential language (3 marks)

d) Define totally new parallel language and compiler system (3 marks)

Question 7 (8 marks; 8 minutes)

Compare the following parallel programming models:

a) Shared Memory Model (2 marks)

b) Threads Model (2 marks)

c) Message Passing Model (2 marks)

d) Data Parallel Model (2 marks)

Question 8 (15 marks; 15 minutes)

Tell whether the following equations are parallelizable or non-parallelizable. Also show how to ***decompose*** the parts of the equations.

a)
```
for (i=1; i<N; i++) {
    a[i] = a[i-1] + a[i-2];
}
```

b)
```
w = a[0] * b[0];
for (i=1; i<N; i++) {
  c[i] = w;
  w = a[i] * b[i];
}
```

c) $F(x) = a*P(x) - b*Q(x) - c*R(x)$

d) $F(m,n) = (m * n)!$

e) $F(x,y,z) =$ square root of $((x+y)^{50} * (y+z)^{100})$

Question 9 (15 marks; 15 minutes)

a) From the following code fragments, explain how the code will be processed according to the *OpenMP* compiler directive. (5 marks)

```
#pragma omp parallel private(i,j)
for (i = 0; i < m; i++) {
  low = a[i];
  high = b[i];
  if (low > high) {
#pragma omp single
    printf ("Exiting (%d)\n", i);
    break;
  }
#pragma omp for
  for (j = low; j < high; j++)
    c[j] = (c[j] - a[i])/b[i];
}
```

b) Apply a proper compiler directive to parallelize the following program. (5 marks)

```
#include <stdio.h>
#include <omp.h>
#define MAX 500000
int main() {
  double ave=0.0, A[MAX]; int i;
  for (i=0;i< MAX; i++) {
    ave + = A[i];
  }
  ave = ave/MAX;
printf("ave = %f\n",ave);
}
```

c) Show the result of the following program and explain the boundary of each variable. (5 marks)

```
#include <stdio.h>
#include <omp.h>
int main() {
   int i= 256; // a shared variable
   #pragma omp parallel
   {
      int x;
      x = omp_get_thread_num();
      printf("x = %d, i = %d\n",x,i);
   }
}
```

Question 10 (15 marks; 15 minutes)

From the following OpenMP and MPI code fragments, 1) explain how the code will be processed and what would be the result for each code fragment 2) if there is something wrong with the code, correct it or suggest better code fragment.

a)
```
if (z < min)
#pragma omp critical
  min = z;
```

b)
```
#pragma omp parallel
{
        #pragma omp section
        w = alpha();
        #pragma omp section
        v = beta();
        #pragma omp section
        y = delta();
}
```

c)
```
MPI_Comm_rank(comm, &rank);
MPI_Comm_size(comm, &np);
If (rank == 0) {
    MPI_Send(msg1, 100, MPI_FLOAT, 1, 1, comm);
    MPI_Send(msg2, 100, MPI_FLOAT, 1, 2, comm);
} else if (rank == 1) {
    MPI_Recv(msg2, 100, MPI_FLOAT, 0, 2, &status, comm);
    MPI_Recv(msg1, 100, MPI_FLOAT, 0, 1, &status, comm);
    Avg = (msg1 + msg2) / 2;
    printf("The average is %f ", Avg);
}
```

----End of Examination----

Pichaya Tandayya Lecturer