PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Midterm Examination: Semester 1

Academic Year: 2013

Date: 3 August 2013

Time: 13.30 - 16.30 (3 hours)

Subject Number: 242-530

Room: Robot

Subject Title: Parallel and Distributed Computing

Exam Duration: 3 hours

This paper has 15 pages, 8 questions and 165 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	55	55		5	15	12	
2	15	12		6	10	8	
3	20	20		7	15	12	
4	20	21		8	20	20	
Total	**170**	**160**			**30%**		

Question 1 (52 marks; 55 minutes)

Answer the following questions.

a) What are disadvantages of *Manual* and *Automatic* Parallelization? (4 marks)

Manual Parallelization	Automatic Parallelization

b) Compare *latency* and *bandwidth.* (2 marks)

Latency	Bandwidth

c) What are the differences between *synchronous* and *asynchronous* communications? (2 marks)

Synchronous Communication	Asynchronous Communication

d) The first step in developing parallel software is to first understand the problem that you wish to solve in parallel. You need to determine whether or not the problem can actually be parallelized. Suppose that you have a serial program and want to parallelize it. What are the *three* necessary things to be *identified* in the program? Also explain them clearly. (3 marks)

e) After finding the *bottlenecks* and *inhibitors* of the program, what can be done to solve the problems? (2 marks)

f) Explain the type of problems called "embarrassingly parallel" and also give examples. (3 marks)

g) List at least 6 factors to consider when designing your program's inter-task communications. (3 marks)

h) 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

i) What is *barrier synchronization*? (2 marks)

j) Explain how *lock* and *semaphore* involve with *synchronization*? (2 marks)

k) Compare *loop independent data dependence* and *loop carried data dependence* by giving an example of code fragment for each type. (2 marks)

Loop independent data dependence	Loop carried data dependence

l) What is *granularity*? (2 marks)

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

Fine-grain Parallelism	Coarse-grain Parallelism

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

o) Give examples of classes of problems that result in load imbalances even if data is evenly distributed among tasks. (3 marks)

p) Why are parallel applications much more complex than corresponding serial applications in general? (2 marks)

q) Explain how the costs of complexity are measured. (4 marks)

r) Explain why more CPU time is required when the primary intent of parallel programming is to decrease the execution wall clock time? (3 marks)

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

Question 2 (12 marks; 15 minutes)

Use Amdahl's law to explain the following questions

a) Why is it that simply adding more machines does not always scale the performance of a parallel program? Give an example. (6 marks)

b) Why does problem size play an important role in parallel programs in terms of performance? Give an example. (6 marks)

____

____

____

____

____

____

____

____

____

____

____

____

____

Question 3 (20 marks; 20 minutes)

From the following statements, tell which are true (T) or false (F).

_____a) Often it is inefficient to package small messages into a larger message, thus decreasing the effective communications bandwidth.

_____b) Communications frequently require some type of synchronization between tasks, which can result in tasks spending time "waiting" instead of doing work.

_____c) Competing communication traffic can saturate the available network bandwidth, further solving performance problems.

_____d) Sending many small messages can cause latency to dominate communication overheads.

_____e) Asynchronous communication operations can decrease overall program performance.

_____f) A data dependence results from multiple use of the same location(s) in storage by different tasks.

_____g) Dependencies are important to parallel programming because they are one of the primary inhibitors to parallelism.

_____h) Coarse-grain parallelism facilitates load balancing.

_____i) If granularity is too fine, it is possible that the overhead required for communications and synchronization between tasks takes shorter than the computation.

_____j) Fine-grain parallelism implies high communication overhead and more opportunity for performance enhancement.

_____k) The most efficient granularity is dependent on the algorithm and the hardware environment in which it runs.

_____l) When the amount of work each task will perform is intentionally variable, or is unable to be predicted, it may be helpful to use a scheduler - task pool approach.

_____m) Periods of computation are typically separated from periods of communication by synchronization events.

_____n) I/O operations are generally regarded as hotspots to parallelism.

_____o) Problems with a fixed percentage of parallel time are more scalable than problems that increase the percentage of parallel time with their size.

_____p) Uniform Memory Access (UMA) requires identical processors.

_____q) Non-Uniform Memory Access (NUMA) provides equal access time to all processors

_____r) The Tree Switched Network Topology performs well if there is a small amount of locality in communication.

_____s) Cache coherent means if one processor updates a location in shared memory, all the other processors know about the update..

_____t) Shared Medium (an interconnection media type) enables multiple messages to be sent simultaneously and allow scaling of network to accommodate increase in processors.

Question 4 (21 marks; 20 minutes)

Answer the following questions.

a) Compare Parallel Vector Processor (PVP) and Symmetric Multiprocessors (SMP). (6 marks)

Parallel Vector Processor	Symmetric Multiprocessors

b) Compare the advantages and disadvantages of Shared Memory Model and Distributed Memory Model. (6 marks)

Shared Memory Model	Distributed Memory Model

c) Compare OpenMP and MPI. (6 marks)

OpenMP	MPI

Name________________________________ID________________

d) List the significant constraints in building faster serial computers. (3 marks)

Question 5 (12 marks; 15 minutes)

Tell the advantages and disadvantages of 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 6 (8 marks; 10 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 7 (12 marks; 15 minutes)

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

a)

```
do i=0,n
   a(i+1) = a(i) * a(i-1)
enddo
```

b)

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

c) $F(x) = a*M(x) + b*N(x) - c*O(x)$

d) $H(i) = F(i)i + G(i)$

e) $F(m,n) = (m+1)! + (n-1)!$

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

Question 8 (20 marks; 20 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)

```
int t_id, num_t;
#pragma omp parallel num_threads(3)
{
        num_t = omp_get_num_threads();
        t_id = omp_get_thread_num();
        printf("Hello world from thread ID %d/%d\n",t_id,num_t);
}
```

c)

```
#pragma omp parallel for
 for (i=0; i<NUMBER; i++)
 {
         result+=(sin(data[i])-cos(data[i]))/tan(data[i]);
 }
 printf("Result = %f\n",result);
```

d)

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

e)

```
MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
if (n != 0)
{
   mysum = 0;
   for (i = process_id + 1; i <= n; i += number_of_processes)
{
         mysum +=i;
      }
MPI_Reduce(&mysum, &sum, 1, MPI_DOUBLE, MPI_SUM, 0,
MPI_COMM_WORLD);
```

____

____

____

____

____

____

____

____

____

____

----End of Examination----

Pichaya Tandayya Lecturer

"It is not the strongest of the species that survive,
nor the most intelligent,
but the one most responsive to change."

By Charles Darwin

Name________________________________ID____________________