

---

UNIVERSITI SAINS MALAYSIA

First Semester Examination  
2011/2012 Academic Session

January 2012

**CST331 – Principles of Parallel & Distributed Programming**  
*[Prinsip Pengaturcaraan Selari & Teragih]*

Duration : 2 hours  
*[Masa : 2 jam]*

---

**INSTRUCTIONS TO CANDIDATE:**

***[ARAHAN KEPADA CALON:]***

- Please ensure that this examination paper contains **FOUR** questions in **EIGHT** printed pages before you begin the examination.

*[Sila pastikan bahawa kertas peperiksaan ini mengandungi **EMPAT** soalan di dalam **LAPAN** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]*

- Answer **ALL** questions. Please write your answers according to the sequence of the questions.

*[Jawab **SEMUA** soalan. Sila tulis jawapan anda mengikut turutan soalan.]*

- You may answer the questions either in English or in Bahasa Malaysia.

*[Anda dibenarkan menjawab soalan sama ada dalam bahasa Inggeris atau bahasa Malaysia.]*

- In the event of any discrepancies, the English version shall be used.

*[Sekiranya terdapat sebarang percanggahan pada soalan peperiksaan, versi bahasa Inggeris hendaklah diguna pakai.]*

---

1. A parallel laboratory has only 4 processors available. Students are given the task of finding the maximum of the following 16 values {78, 34, 67, 21, 123, 87, 243, 9, 17, 74, 98, 21, 87, 23, 450, 76} using these processors.
- (a) What is the most suitable decomposition method for the problem? Why?  
(4/100)
- (b) Explain how this problem may be solved as a parallel program using an appropriate parallel algorithm model. Accompany your answer with appropriate illustration.  
(10/100)
- (c) If the laboratory received another 4 new processors, how would you decompose the task to be run on all 8 processors?  
(6/100)
- (d) If 70% of the sequential code for finding maximum is suitable for parallelization, what is the speed up for a parallel version of this code using 8 processors? Calculate using Amdahl's and Gustafson-Barsis's law.  
(6/100)
- (e) Discuss why you got two different values in Question 1(d) above. Give **two (2)** reasons.  
(4/100)
2. (a) Explain cache coherence problem and state its **two (2)** main solutions.  
(6/100)
- (b) Chip Multi-processor (CMP), Cluster and Symmetric multiprocessor (SMP) are parallel machines. Explain each of them based on Flynn's taxonomy, parallel programming model, granularity and system architecture.  
(12/100)
- (c) What does the following align and distribution mechanism in High Performance Fortran (HPF) do? You may explain by using an example.
- (i) !HPF\$ ALIGN
- (ii) !HPF\$ DISTRIBUTE (BLOCK)
- (iii) !HPF\$ DISTRIBUTE (CYCLIC)  
(6/100)

3. (a) OpenMP is based on the fork/join parallelism which is the standard view of shared-memory model. Explain how the fork/join parallelism works.

(10/100)

- (b) Given the code below, answer the following questions:

```
#pragma omp parallel private (t,x,y,local_count)
{
  local_count=0;
  Random^rand=gcnew Random();
  t=omp_get_num_threads();

  #pragma omp parallel for
  for (int i=tid; i<samples; i+=t)
  {
    x=rand->Next(0,1000)*0.0001;
    y=rand->Next(0,1000)*0.0001;
    if(x*x+y*y<=1.0) local_count++;
  }

  #pragma omp critical
  count+=local_count;
}
```

- (i) What is the role played by the clause *#pragma omp parallel for*?
- (ii) What are the local variables in each thread and in which clause they are defined?
- (iii) The code given has a critical section denoted by the *#pragma omp critical*. What does critical section means and why it is needed in this code?

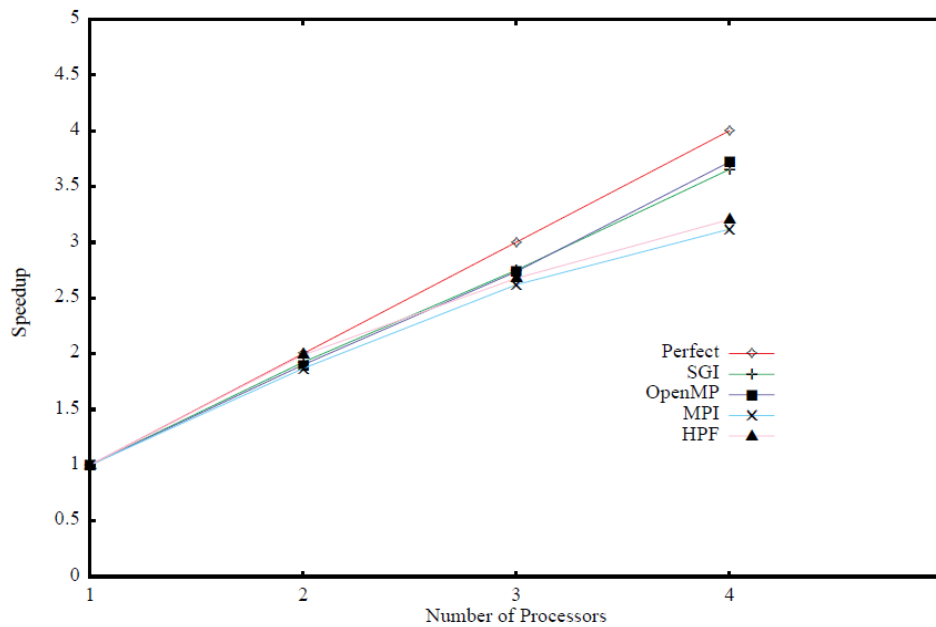
(12/100)

- (c) Give the parallel version of the following code using OpenMP:

```
float array_sum (float a[], int n)
{
  int i;
  float sum=0.0;
  for (i=0; i<n; i++)
    sum+=a[i];
  return (sum);
}
```

(4/100)

4. (a) The following plot describes the parallel performance for the 250 x 250 game-of-life grid. The line with diamond shapes indicates a perfect case for parallelisation. However, the performance shown by three main parallel programming model (i.e. OpenMP, MPI and HPF) do not reach the perfect case. Define perfect (linear) case and give two reasons why there are gaps between the perfect case and the other parallel programming models.



(10/100)

- (b) Given the following table on the timings (in seconds) for the game-of-life with gridsizes of 250 X 250, discuss the timing of each programming model related to their theoretical concept. You should also consider the trend observed from the table in relation to the number of processors, etc.

Number of processors	1	2	3	4
Programming Model				
OpenMP	79.6	41.9	29.1	21.4
MPI	93.2	49.9	35.6	29.9

(10/100)

1. Sebuah makmal komputer selari mempunyai hanya 4 buah pemproses. Para pelajar diberi tugas untuk mencari maksimum daripada 14 nilai berikut {78, 34, 67, 21, 123, 87, 243, 9, 17, 74, 98, 21, 87, 23, 450, 76} menggunakan pemproses yang sedia ada.
  - (a) Apakah kaedah pembahagian yang paling sesuai untuk masalah ini? Kenapa?

(4/100)
  - (b) Terangkan bagaimana masalah ini boleh diselesaikan sebagai program selari menggunakan model algoritma selari yang sepatutnya.

(10/100)
  - (c) Jika makmal tersebut menerima 4 lagi pemproses baru, bagaimanakah anda akan membahagikan tugas ini untuk dilaksanakan menggunakan 8 pemproses?

(6/100)
  - (d) Jika hanya 70% daripada kod jujukan untuk mencari maksimum ini sesuai untuk dipinda kepada kod selari, apakah tahap "speed-up" masa larian yang boleh dicapai jika ia dilaksanakan menggunakan 8 pemproses? Buat pengiraan menggunakan hukum Amdahl dan Gustafson-Barsis.

(6/100)
  - (e) Bincang kenapa anda mendapat dua jawapan yang berbeza bagi Soalan 1(d) di atas. Berikan **dua (2)** sebab.

(4/100)
  
2.
  - (a) Terangkan masalah 'cache-coherence' dan nyatakan **dua (2)** cara menyelesaikannya.

(6/100)
  - (b) Chip multi-processor (CMP), Kluster dan Symmetric Multi-processor (SMP) adalah mesin-mesin selari. Terangkan setiap satu berdasarkan taksonomi Flynn, model pengaturcaraan selari, granulariti dan reka bentuk sistem.

(12/100)

(c) Apakah yang dilakukan oleh mekanisme 'Align' dan 'Distribution' dalam High Performance Fortran (HPF) berikut. Anda boleh jelaskan dengan menggunakan contoh.

- (i) !HPF\$ ALIGN
- (ii) !HPF\$ DISTRIBUTE (BLOCK)
- (iii) !HPF\$ DISTRIBUTE (CYCLIC)

(6/100)

3. (a) OpenMP adalah berdasarkan perselarian 'fork/join' iaitu pandangan piawai dalam model memori kongsi. Terangkan bagaimana perselarian 'fork/join' ini berfungsi.

(10/100)

(b) Diberi atur cara seperti di bawah, jawab soalan-soalan berikut:

```
#pragma omp parallel private (t,x,y,local_count)
{
local_count=0;
Random^rand=gnew Random();
t=omp_get_num_threads();

#pragma omp parallel for
for (int i=tid; i<samples; i+=t)
{
x=rand->Next(0,1000)*0.0001;
y=rand->Next(0,1000)*0.0001;
if(x*x+y*y<=1.0) local_count++;
}

#pragma omp critical
count+=local_count;
}
```

- (i) Apakah peranan yang dimainkan oleh klausa *#pragma omp parallel for*?
- (ii) Apakah pemboleh ubah tempatan bagi setiap bebenang dan di dalam klausa yang manakah mereka diisytiharkan?
- (iii) Atur cara yang diberikan di atas mempunyai bahagian kritikal yang ditandakan oleh klausa *#pragma omp critical*, apakah yang dimaksudkan dengan bahagian kritikal dan kenapa ia diperlukan dalam atur cara tersebut?

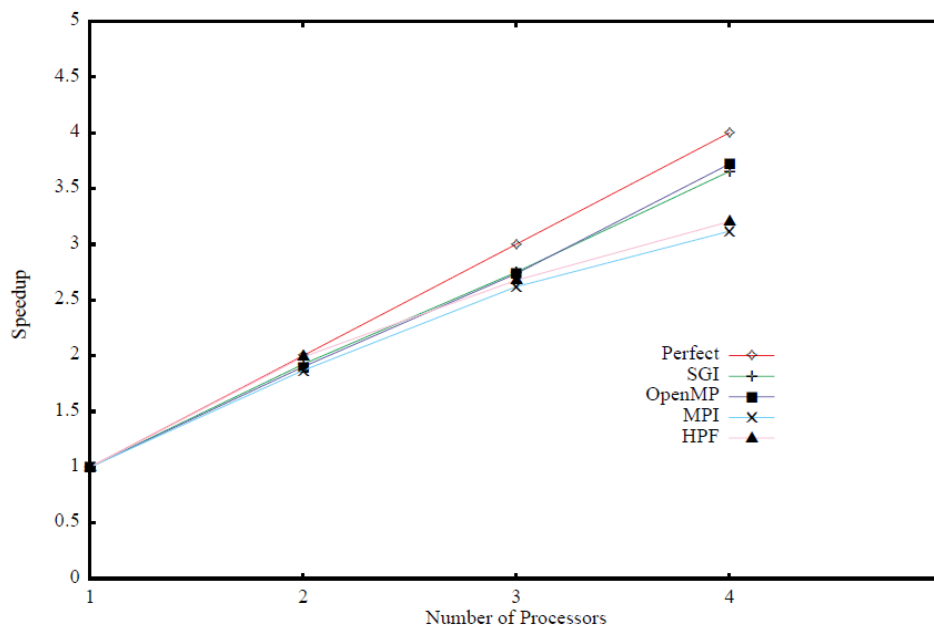
(12/100)

- (c) Berikan versi selari bagi atur cara berikut menggunakan OpenMP:

```
float array_sum (float a[], int n)
{
    int i;
    float sum=0.0;
    for (i=0; i<n; i++)
        sum+=a[i];
    return (sum);
}
```

(4/100)

4. (a) Plot berikut menunjukkan prestasi selari untuk grid 250 X 250 "game-of-life". Garisan dengan bentuk berlian menggambarkan kes sempurna bagi perselarian. Namun demikian, prestasi yang ditunjukkan oleh tiga model pengaturcaraan selari yang utama (iaitu OpenMP, MPI and HPF) tidak mencapai tahap kes sempurna (linear). Definisikan kes sempurna dan bincang kenapa wujud jurang antara kes sempurna dengan kes model pengaturcaraan selari yang lain.



(10/100)

- (b) Diberikan jadual di bawah yang memaparkan jangkamasa (dalam saat) untuk larian "game-of-life" menggunakan grid 250 X 250, bincang jangkamasa larian untuk setiap model pengaturcaraan dengan mengaitkannya kepada konsep teori masing-masing. Anda juga patut mempertimbangkan corak yang diperhatikan daripada jadual dari segi bilangan pemproses dan lain-lain.

Bilangan Pemproses	1	2	3	4
Model Pengaturcaraan				
OpenMP	79.6	41.9	29.1	21.4
MPI	93.2	49.9	35.6	29.9

(10/100)