**National University of Computer & Emerging Sciences, Karachi  
 Fall -2021 CS-Department  
Mid-Term 1 Exam   
15th October 2021, 10:30 am – 11:30 am**Fast

|  |  |  |
| --- | --- | --- |
| **Course Code:** CS3006 | **Course Name:** Parallel and Distributed Computing | |
| **Instructors Names:** Dr. Nadeem Kafi Khan, Dr. Hassan Jamil Syed, Dr. Nausheen Shoaib and Mr. Muhammad Danish Khan | | |
| **Student Roll No:** | | **Section No:** |

Instructions:

* Return the question paper.
* Read each question completely before answering it. Understanding the questions is part of the examination. There are **3 questions and 2 pages.**
* All the answers must be solved according to the sequence given in the question paper.
* Examples are mandatory, where asked.
* Illustrate means drawing a diagram and labelling it in detail only. Textual details are not required.

Time: 60 minutes. Max Marks: 30

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Question No. 1. Short questions, each part has 2 Marks. [10 Marks]

1. How is serial computing different from parallel computing? What compute resources are required for parallel computing?

Ans:  
Traditionally, software has been written for ***serial*** computation:

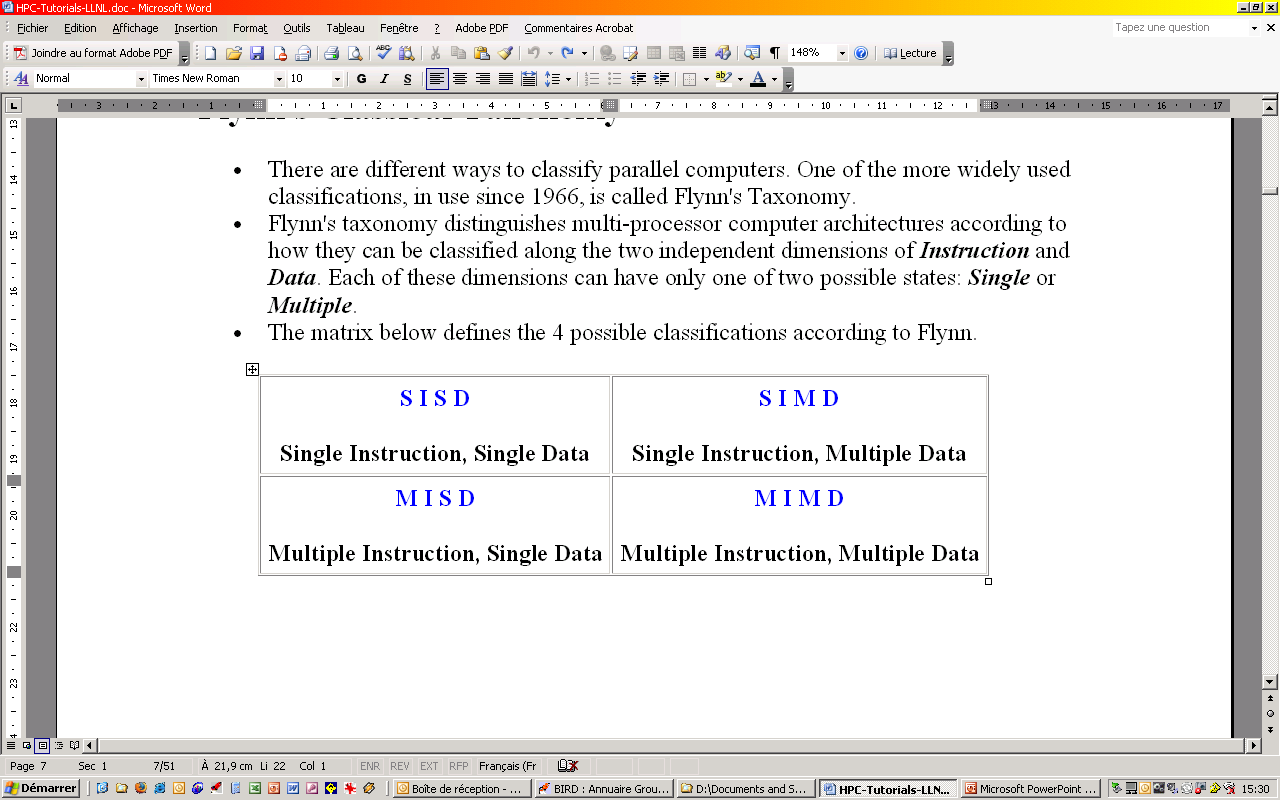
* + To be run on a single computer having a single Central Processing Unit (CPU);
  + A problem is broken into a discrete series of instructions.
  + Instructions are executed one after another.

Only one instruction may execute at any moment in time.

* In the simplest sense, ***parallel computing*** is the simultaneous use of multiple compute resources to solve a computational problem.
  + To be run using multiple CPUs
  + A problem is broken into discrete parts that can be solved concurrently
  + Each part is further broken down to a series of instructions
* Instructions from each part execute simultaneously on different CPUs
* The compute resources can include:
  + A single computer with multiple processors;
  + A single computer with (multiple) processor(s) and some specialized computer resources (GPU, FPGA …)
  + An arbitrary number of computers connected by a network;
  + A combination of both.

1. Differentiate between fine-grained and coarse-grained computing models through an example.
2. How does Flynn's taxonomy distinguish between multi-processor computer architectures?
3. Ans:

The matrix below defines the 4 possible classifications according to Flynn



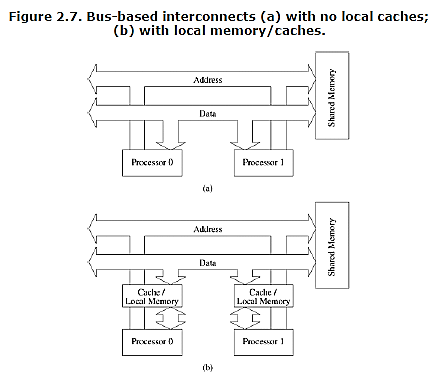
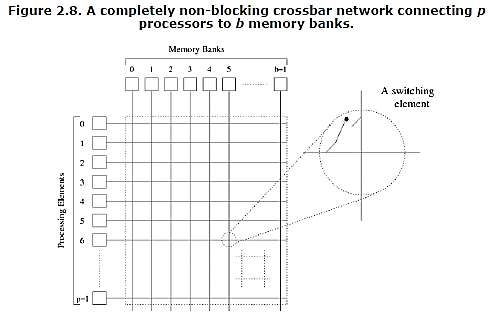
1. How is scalability important in parallel computing? Explain using an example.
   1. Refers to a parallel system's (hardware and/or software) ability to demonstrate a proportionate increase in parallel speedup with the addition of more processors. Factors that contribute to scalability include:
      1. Hardware - particularly memory-cpu bandwidths and network communications
      2. Application algorithm
      3. Parallel overhead related
      4. Characteristics of your specific application and coding
2. What are the advantages and disadvantages of Distributed Shared Memory Architecture?
   1. Adv. Most programmer are well-versed in shared memory programming model, therefore, low learning curve.
   2. Adv. Applications written for shared memory systems can run unchanged on distributed memory architecture.
   3. DisAdv. As the underlying infrastructures is distributed, the created single address space abstraction will be NUMA type.

Question No. 2

1. Illustrate how memory hierarchy can affect system performance. Clearly label the diagram to show how performance can be improved. [5 Marks]

Student should have drawn processor-cache-main memory diagram. S/he also need to specific that cache will reduce latency (thus increasing performance). Label could be cache hits, cache misses, spatial and temporal locality.

1. Illustrate bus and crossbar interconnects between processors and memory. How are crossbars more beneficial than buses? Explain using an example. [5 Marks]



Crossbar is non-blocking network in the sense that the connection of a processing node to a memory bank does not block the connection of any other processing nodes to other memory banks.   
*\*The student should have identified two different non-block but concurrent paths in the crossbar diagram.*

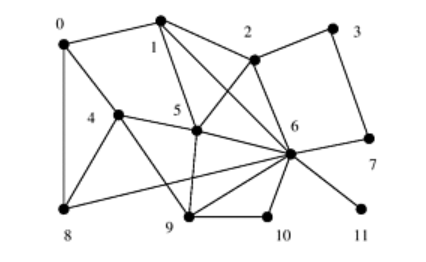
Question No. 3 [5 + 5 Marks]

1. Consider the problem of multiplying a sparse matrix “*A”* with a vector “*b”* (See Figure 1). Unlike a dense matrix-vector product though, only non-zero elements of matrix A participate in the computation.

**Draw the corresponding task-interaction graph** of dot product of sparse matrix “*A”* with the vector “*b”*. Make suitable assumptions if required. *Note. There is no need to show multiplication operations.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | **A** |  |  |  |  |  |  |  | **b** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* |  |  |
| *0* | **7** | **9** | **0** | **0** | **5** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |  | **6** |
| *1* | **9** | **2** | **3** | **0** | **0** | **8** | **3** | **0** | **0** | **0** | **0** | **0** |  | **2** |
| *2* | **0** | **4** | **5** | **2** | **0** | **9** | **5** | **0** | **0** | **0** | **0** | **0** |  | **1** |
| *3* | **0** | **0** | **2** | **4** | **0** | **0** | **0** | **7** | **0** | **0** | **0** | **0** |  | **7** |
| *4* | **7** | **0** | **0** | **0** | **3** | **3** | **0** | **0** | **3** | **7** | **0** | **0** |  | **3** |
| *5* | **0** | **6** | **4** | **0** | **6** | **1** | **5** | **0** | **0** | **4** | **0** | **0** |  | **8** |
| *6* | **0** | **2** | **7** | **0** | **0** | **5** | **1** | **2** | **0** | **9** | **8** | **5** |  | **7** |
| *7* | **0** | **0** | **0** | **1** | **0** | **0** | **4** | **9** | **0** | **0** | **0** | **0** |  | **9** |
| *8* | **5** | **0** | **0** | **0** | **3** | **0** | **3** | **0** | **7** | **0** | **0** | **0** |  | **2** |
| *9* | **0** | **0** | **0** | **0** | **9** | **5** | **3** | **0** | **0** | **3** | **7** | **0** |  | **3** |
| *10* | **0** | **0** | **0** | **0** | **0** | **0** | **7** | **0** | **0** | **2** | **3** | **0** |  | **4** |
| *11* | **0** | **0** | **0** | **0** | **0** | **0** | **3** | **0** | **0** | **0** | **0** | **2** |  | **3** |

Figure 1. Matrix A and Vector b



1. Explain how the task-interaction graph helps improve performance.

Vector b will have a single copy. Different rows of the matrix assigned to different processes running in parallel can either store the entire copy or access only elements needed in case of sparse matrix computation (non-zero). Task interactions graphs can be used to retrieve this information. Reading selective part of b is important as all values of b cannot fit in a cache line for large vectors.

THE END