Assignment #1

CS4379: Parallel and Concurrent Programming
CS5379: Parallel Processing
Spring 2020

Due Date: 2/6, 12:30 p.m., please submit a soft copy via Blackboard (preferred) or a hard copy in class. Late submissions are accepted till 2/11, 12:30 p.m., with 10% penalty each day. No submissions accepted after 2/11, 12:30 p.m.

Please name your submission file starting as "LastName_FirstName_HW1".

Q1. Please go to the Top 500 Supercomputers site (http://www.top500.org/) and list the **ten most powerful supercomputers** along with their FLOPS rating.

ANSWER: Information is retrieved via the following link: https://www.top500.org/lists/2019/11/

No	**Supercomputers**	**Linpack Performance (Rmax)**	**Theoretical Peak (Rpeak)**
1	Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband	148,600 TFlop/s	200,795 TFlop/s
2	Sierra - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband	94,640 TFlop/s	125,712 TFlop/s
3	Sunway TaihuLight - Sunway	93,014.6 TFlop/s	125,436 TFlop/s

	MPP, Sunway SW26010 260C 1.45GHz, Sunway		
4	Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000	61,444.5 TFlop/s	100,679 TFlop/s
5	Frontera - Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox InfiniBand HDR	23,516.4 TFlop/s	38,745.9 TFlop/s
6	Piz Daint - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100	21,230 TFlop/s	27,154.3 TFlop/s
7	Trinity - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect	20,158.7 TFlop/s	41,461.2 TFlop/s
8	AI Bridging Cloud Infrastructure (ABCI) - PRIMERGY CX2570 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR	19,880 TFlop/s	32,576.6 TFlop/s
9	SuperMUC-NG - ThinkSystem SD650, Xeon Platinum 8174 24C 3.1GHz, Intel Omni-Path	19,476.6 TFlop/s	26,873.9 TFlop/s
10	Lassen - IBM Power System AC922, IBM POWER9 22C 3.1GHz, Dual-rail Mellanox EDR Infiniband, NVIDIA Tesla V100	18,200 TFlop/s	23,047.2 TFlop/s

Q2. Please list two examples of “parallel processing” you can think of in daily life.

ANSWER:

1. Example 1: Walmart checkout can be considered as parallel processing where each counter is a processing unit, each customer is an input.
2. Example 2: Lubbock Driver License is another daily life example that illustrates parallel processing. Upon getting a ticket, a customer will be directed to an officer whose previous job is done first. This type of parallel can be considered as a single stream, multiple processors

Q3. Please list two reasons you would argue for having parallel processing.

ANSWER:

1. First parallel processing would significantly increase SPEED in performing unrelated tasks like examples in Q2. In this regard, each processing unit can work independently without affecting the reliability of the whole system
2. Second, parallel processing allows us to maximize utilizing CPUs use. According to Moore's law, CPU power doubles every 18 months, higher than that of memory speed (approximately 10%). Without parallel processing, there could be a waste in CPU utilization since the speed of CPU operation is faster than accessing memory (e.g., CPU has to wait for feeding the data from memory)

Q4. Please briefly explain the following acronyms/terminologies we discussed in class regarding parallel computer architectures: **SISD, SIMD, MISD, MIMD, and SPMD.**

ANSWER

1. **SISD** (Single Instruction, Single Data): This is the model of serial Von Neumann machine where only one instruction stream is executed by the CPU during any clock cycle. In addition, only one data stream is fed as an input during any one clock cycle
2. **SIMD** (Single Instruction, Multiple Data) is an architecture where all processing units will execute the same instruction at any given clock cycle. Furthermore, each processing unit will operate on a different dataset element.

3. **MISD** (Multiple Instruction, Single Data). In this architecture, a single data stream is fed into multiple processing units and each processing unit will operate on the data independently through independent instruction streams
4. **MIMD** (Multiple Instruction, Multiple Data) is the most prevalent type of parallel computer where each processing unit may execute different instruction streams and different data streams
5. **SPMD** (Single Program, Multiple Data) is a variant of MIMD where the same program (multiple instances are executed on different processing units. Every processor can work on multiple data streams

Q5. Please briefly explain the following acronyms/terminologies we discussed in class regarding parallel computer architectures: **UMA, NUMA, ccNUMA, and DSM.**

ANSWER

1. **UMA** (Uniform Memory Access): is an architecture used in parallel computing where all processors share the physical memory uniformly. It is noted that the time taken to access a memory address is independent of requesting processors.
2. **NUMA** (Non-Uniform Memory Access): In contrast to UMA, this memory design poses a dependence of the memory access time with the memory location relative to the processor.
3. **ccNUMA** (Cache coherent NUMA) uses a fast non-shared memory to exploit locality of reference in memory accesses
4. **DSM** (Distributed shared memory) is an architecture where physically separated memories can be accessed as one logically shared address space.

Q6. What are the advantages and disadvantages of shared address space and distributed address space parallel computers, respectively?

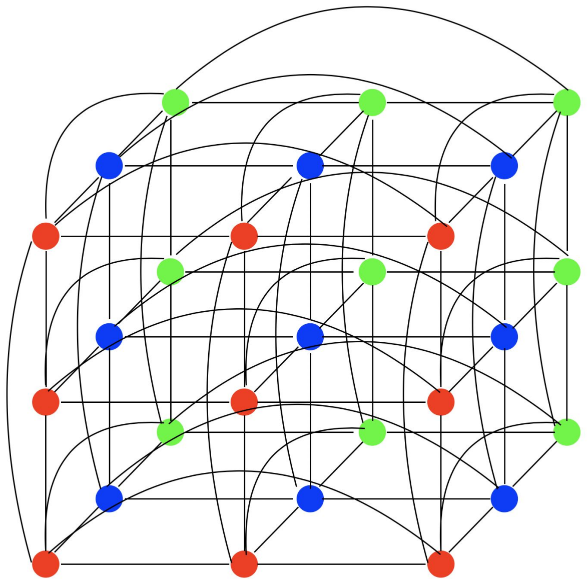
ANSWER:

	Advantages	Disadvantages
Shared Address	● Global address space	● Need hardware/software

Space	view ● Easier to program ● Implicit communication	to support view ● Complexity in system design ● Not easy to be scalable
Distributed Address Space	● Requires little hardware support, other than a network ● Easy to scale up	● No global address space view ● More difficult to program ● Need explicit communication

Q7. Please draw a diagram to show what the 3-D torus interconnection topology with 27 processors looks like.

ANSWER:



Q8. A cycle in a graph is defined as a path originating and terminating at the same node. The length of a cycle is the number of edges in the cycle. Show that **there are no odd-length cycles in a *d*-dimensional hypercube**.

ANSWER:

We can use *d*-dimensional bits to represent each node in the graph. It is noted that when traversing through the hypercube network each node is different exact one bit. Based on the formation of the hypercube we can encode each node as ODD if the total number of bit 1 is ODD or even otherwise. For example:
0001 => ODD
0101 => EVEN
To form a cycle in a graph, a node must start and end with the same representation (both ODDs or EVENs). For example, a node sends a message to its neighbour and traverses back will have the form

ODD - EVEN- ODD or EVEN - ODD - EVEN where each "-" represents a *path*

Using automata approach we can generalize the paths as:

ODD-EVEN (-OOD - EVEN)*- ODD or EVEN - ODD (-EVEN - ODD)* - EVEN
Where * is the intermediate node and >= 0 and "-" represents a *path*
In this case the length of the cycle is calculated as: 2 + 2* which is an **EVEN** number or there is no odd length cycles.
Hint: consider how a *d*-dimensional hypercube is constructed, and how the binary number of the processor label changes when traversing through the hypercube network.

Q9. A ***mesh of trees*** is a network that imposes a tree interconnection on a grid of processing nodes. A $\sqrt{p} \times \sqrt{p}$ mesh of trees is constructed as follows. Starting with a $\sqrt{p} \times \sqrt{p}$ grid, a complete binary tree is imposed on each row of the grid. Then a complete binary tree is imposed on each column of the grid. Figure 1 illustrates the construction of a 4 x 4 mesh of trees. Assume that the nodes at intermediate levels are switching nodes. Please determine: 1) the bisection width, 2) the diameter, and 3) the total number of switching nodes in a $\sqrt{p} \times \sqrt{p}$ mesh.

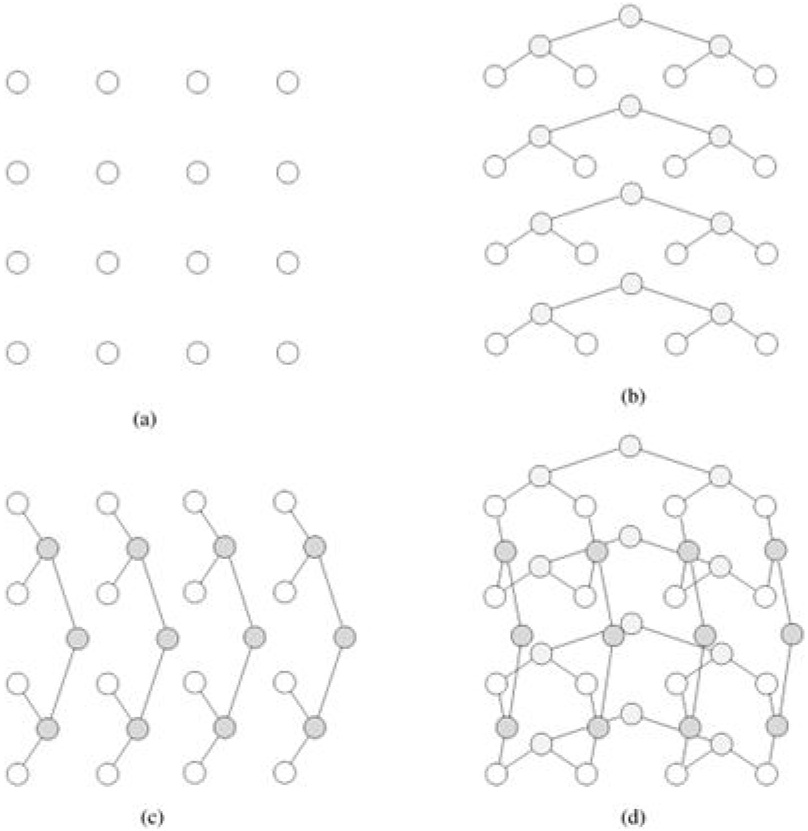


Figure 1. The construction of a 4 x 4 mesh of trees: (a) a 4 x 4 grid, (b) complete binary trees imposed over individual rows, (c) complete binary trees imposed over each column, and (d) the complete 4 x 4 mesh of trees.

ANSWER:

1. Bisection width**:** is defined as the minimum number of links needed to be removed to partition a network. Since the network is constructed by combining horizontal binary trees and vertical binary trees and none of the vertical or horizontal trees are connected locally. We can perform the cut on either direction which is $\sqrt{p}$
2. The diameter is defined as the maximum distance between any two nodes which are the two other end nodes lying on the diagonal of the grid. To travel between these two nodes, the message has to traverse completely on the vertical binary tree then horizontal binary tree or vice versa.
   The total number of nodes in a binary tree is: 2 $\sqrt{p}$ -1
   The maximum cost to travel between two nodes is:

$2 \log ((2\sqrt{p}-1 + 1)/2) = 2 \log (2\sqrt{p}/2) = 2 \log \sqrt{p}$

The cost to travel on each binary tree is: $2 \log \sqrt{p}$

Therefore the max total cost (or the diameter) is $2*2 \log \sqrt{p} = 4 \log \sqrt{p}$

3. Total number of switching nodes:
    a. Total of nodes in one binary tree: $2\sqrt{p}-1$
    b. Total of switching nodes on one binary tree: $2\sqrt{p}-1 - \sqrt{p} = \sqrt{p} -1$
    c. We have $\sqrt{p}$ trees on each dimension
    d. So the total of switching nodes is:
    $\sqrt{p}\ (\sqrt{p} -1) + \sqrt{p}\ (\sqrt{p} -1) = 2 * \sqrt{p}\ (\sqrt{p} -1)$

THE END.