# EE/CS 451 Midterm 2

# Open book, Open notes

Fall 2020

Instructor: Viktor Prasanna

3:30 – 5:30pm

Friday, 10/23/2020

| Problem # | Topic | Points | Score |
|---|---|---|---|
| 1 | Scalability | 15 | |
| 2 | Communication Primitives | 15 | |
| 3 | Communication Primitives | 15 | |
| 4 | CUDA/GPU | 10 | |
| 5 | Task Dependency Graph | 10 | |
| 6 | Data Decomposition | 15 | |
| Total | | 100 | |

Student Name:

Student USC-ID:

The exam duration is 120 mins (3:30pm-5:30pm). You have 5 mins to download the exam in advance (Piazza->Resources->Exams), and 10 mins to upload the completed exam at the end (Blackboard->Assignments->Midterm 1).

- **Option 1**: Download the exam electronically, use a writable tablet for writing your answers, stop writing at 5:30pm, export the exam to PDF and upload it before 5:40pm.
- **Option 2**: Download the exam, write clearly on paper with problem numbers labeled at the top of each page, stop writing at 5:30pm, scan the paper and upload before 5:40.

Your writing must be **recognizable with human eyes**. Any un-clear answer may not receive full credit.

Please turn on your camera on Zoom.

If you have any questions regarding the exam, use the chat on Zoom.

Name Initials: _____

**Problem 2 (15 points)**
**Scalability:**

Assume $n$ processors are connected using a hypercube in the network model.

1. Design a parallel algorithm to find the sum of *n* numbers in $O(\log n)$ time. Assume that initially each processing element is assigned one of the numbers to be added. What is the speed-up? What is the efficiency? Use order notation for speedup and efficiency. (3 + 2 + 2 = 7 points)

2. Design an algorithm using $p < n$ processors. Assume that initially each processing element is assigned $n/p$ numbers to be added. Specify the range of *n* such that efficiency is constant. (4 + 4 = 8 points)

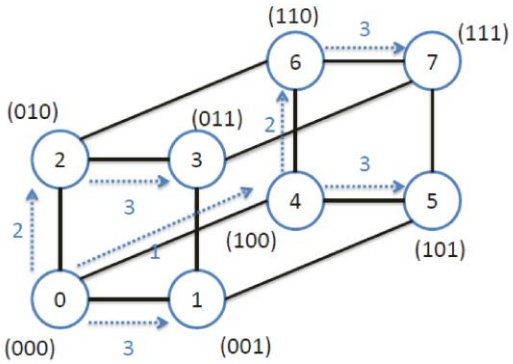
Name Initials: _____

**Problem 3 (15 points)**

**Communication Primitives:**

In the class, we discussed a hypercube One-to-all-broadcast algorithm in which the communication starts with the highest dimension (the dimension specified by the most significant bit of the binary representation of a node index) and proceeds along successively lower dimensions. The following figure illustrates it on an eight-node hypercube with node 0 as the source.

1. Write an iterative pseudo code for One-to-all broadcast on a $p$-node ($p = 2^k$) hypercube (network model) where the communications start with the lowest dimension. Assume the data to be broadcast is 1 unit. (6 points)




2. What is the running time of the algorithm? You may use order notation. (4 points)

3. Show how the algorithm can be simulated in a 1-D mesh of size $p$ to run efficiently. What is the running time? (5 points)

Name Initials: _____

**Problem 4 (15 points)**
**Communication Primitives:**
Suppose we want to do Gather operation on a hypercube.

1. Write a recursive program to perform the Gather operation in a $p$-node ($p = 2^k$) hypercube with node 0 as the destination using the network model. Assume the data to be gathered is of size 1 unit in each node. (5 points)

2. Derive an exact expression for the total number of steps (parallel recursive calls) and the total communication time. (5 points)

3. If each node in the hypercube can use multiple ports to send and receive, what is the lower bound for the total communication time of Gather operation? What is the lower bound if each node uses single port communication? Explain. (5 points)

Name Initials: _____

**Problem 5 (10 points)**

**CUDA GPU:**

Design a CUDA program to perform matrix multiplication $C = A \times B$. The size of each matrix is 1K × 1K. Each element is 1 byte. The matrices are initially stored in the global memory of GPU. The GPU has one streaming multiprocessor (SM) with 1K CUDA cores. Each CUDA core runs at 1GHz and can perform one floating point operation in each clock cycle. The peak bandwidth between the GPU and the global memory is 100 GB/s and the cache of the GPU has been disabled.

1. Design a simple CUDA program using straightforward vector inner product approach. Write the pseudo code for your CUDA program including both kernel function and host function. (3 points)
   Derive a lower bound on the execution time of your kernel function. Explain. (2 points)

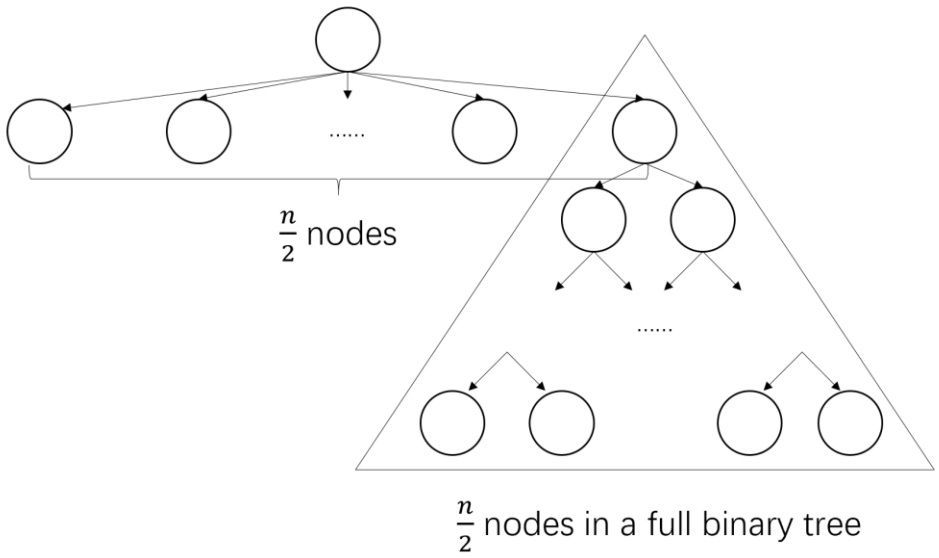
2. Assume the GPU has an on-chip buffer (shared memory) which can store 3 × 32 × 32 elements and has a peak access bandwidth of 1 TB/s.
   Write pseudo code for an optimized CUDA program using block matrix multiplication approach. Show the kernel function and the host function. (2 points)
   Derive lower bounds for 1) the computation time, 2) the local data access time in SM, and 3) the global data access time of your optimized kernel function. Explain. (3 points)

Name Initials: _____

**Problem 6 (10 points)**

**Task Dependency Graph:**

The task dependency graph shown below has $n$ tasks ($n/2 = 2^k - 1$). Assume each task is of unit weight. Determine the following:




1. Maximum degree of concurrency (4 points)

2. Critical path length (2 points)

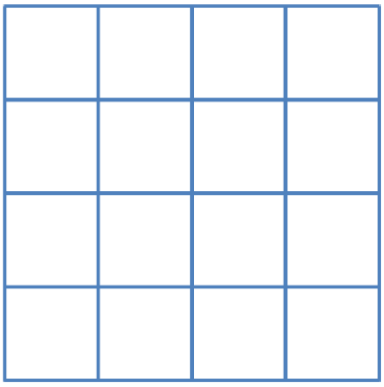
3. The maximum achievable speedup over single processor when the number of processors $p = n$. (4 points)

Name Initials: _____

**Problem 7 (15 points)**
**Data Decomposition:**
Assume an $n \times n$ matrix is partitioned using block cyclic distribution with each block of size 1 and mapped to $p$ processes (assume $n$ and $p$ are powers of 2, $p \leq n$). Assume row major order.

1. For $n$ = 4, $p$ = 2, draw a data partitioning diagram showing the assignment of the elements to processes. (3 points)

Partitioning diagram:



2. Define the mapping function showing how the elements are assigned to each process as a function of the index of the elements, *n* and *p*. (4 points)

3. Consider $n \times n$ matrix multiplication $C = A \times B$. Assume $p = n$. Partition the input and output matrices among $p$ processes using the approach above. Assume the owner of $C(i,j)$ computes $C(i,j)$. What is the total amount of data communication (among all the processes)? (8 points)

Blank Sheet

Blank Sheet