## CSCI-UA.0480-003

# Parallel Computing

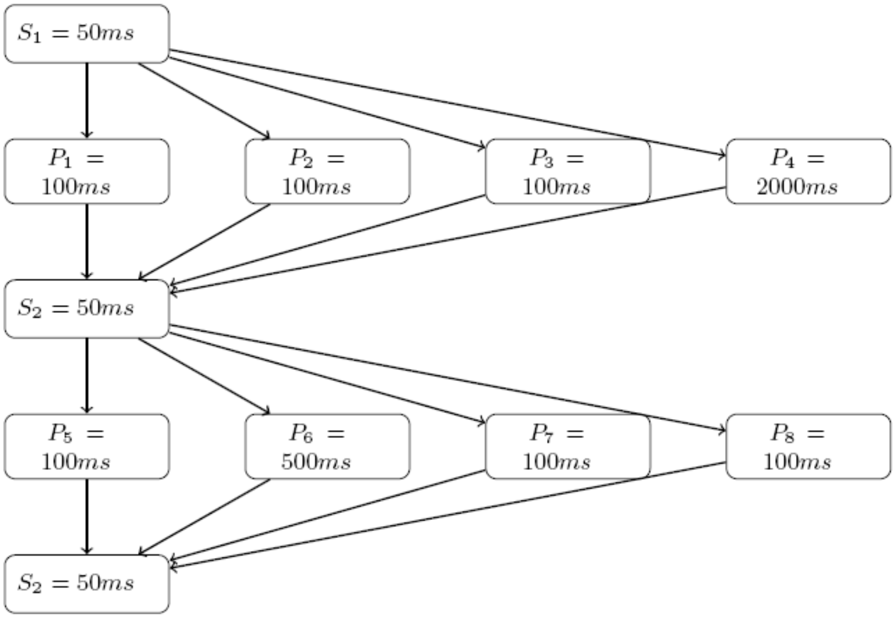
## Midterm Exam

## Spring 2017 (60 minutes)

**NAME:** NetID:

- This exam contains **9 questions** with a total of 40 points in **5 pages.**
- If you have to make assumptions to continue solving a problem, state your assumptions clearly.

1. Assume we have the following task graph. A task can be thought of as function/procedure. Every task is labeled with its run time on a core. An arrow from a task to another means that the first task generates data needed by the second one. Assume all data are of the same size. For all the questions of this problem, assume we have enough cores to run each process on a separate core.



a. [3 points] what is the *best* number of processes to parallelize that program? Justify.

b. [3 points] What will be the speedup if we have 2 processes? 4 processes? 8 processes?

c. [2 points] What is the span for the above graph? what is the work?

d. [1 point] What is the parallelism?

e. [1 point] What does the number you calculated in d mean?

2. [6 points] State 3 shortcomings of Amdahl's law.

-
-
-

3. [6 points] State three factors that can negatively affect the scalability (as we keep adding processes) of a parallel program.

    -
    -
    -

4. [3 points] Can a process in MPI send a pointer to another process? If yes, write the lines for the send and receive between the two processes. It no, justify.

5. [3 points] Does coherence protocols cause extra communication overhead in MPI? Justify.

6. [3 points] Suppose that you want to synchronize (i.e. put a barrier) among several processes but *without using MPI_Barrier()*. What can you do?

7. [2 points] How does MPI programs deal with race conditions?

8. [4 points] State 2 reasons why two processes assigned to two different cores (and no other processes share the cores with them) and executing the same piece of code may not finish at the same time.

-

-

9. [3 points] Suppose a process wants to send data to a subset of processes in MPI. There are two options: (a) several send-receive pairs between this process and each one of the destinations. (b) Split the communicator then use a collective call to send the data to the destinations. Which option do you think is faster? and why?