CS546 "Parallel and Distributed Processing"
Homework 3

Submission:
Due by 11:59pm of 10/15/2019
Total points 130 - Late penalty: 10% penalty for each day late
Please upload your assignment on Blackboard with the following name:
CS546_SectionNumber_LastName_FirstName_HW3.
Please do NOT email your assignment to the instructor and/or TA!

1. **(15 points)** Your company has bought a new 8-core processor, and you have been asked to optimize your software for this processor. You will run two applications on this 8-core processor, but the resource requirements are not equal. The first application needs 80% of the resources, and the other only 20% of the resources.
   a) Given that 40% of the first application is parallelizable, how much speedup would you achieve with at application if run in isolation?
   b) Given that 99% of the second application is parallelizable, how much speedup would this application observe if run in isolation?
   c) Given that 40% of the first application is parallelizable, how much overall system speedup would you observe if you parallelized it?
   d) Given that 99% of the second application is parallelizable, how much overall system speedup would you get?
   e) If we follow fixed-time scalable up principle, and assume only parallelizable portion will scale up in size, what is the fixed-time speedup of question a) and b), respectively?
   f) If we further assume that the applications are dense matrix computations, that is the memory requirement increases with n^2 and computation increases with n^3. Repeat e) for memory bounded speedup.

2. **(10 points)** Prove Amdahl's law (i.e., when p goes to infinite, under Amdahl's law the speedup is ..?).

3. **(10 points)** Assume that we make an enhancement to a computer that improves some mode of execution by a factor of 10. Enhanced mode is used 50% of the time, measured as percentage of the execution time when the enhanced mode is in use. Recall that Amdahl's Law depends on the fractions of the original, non-enhanced execution time that could make use of enhanced mode. Thus, we cannot directly use this 50% measurement to compute speedup with Amdahl's Law.
   a) What is the speedup we have obtained from fast mode?
   b) What percentage of the original execution time has been converted to fast mode?

4. **(10 points)** Calculate the speedup of the three models of parallel speedup assuming the fraction that cannot be parallelized is a = 0.2, and the number processors is p = 20.
   a) Amdahl's law
   b) Gustafson's law
   c) Sun/Ni's law, assuming G(10)=5

5. **(5 points)** What are the four steps to design a parallel algorithm? What are the five methods/patterns of parallel algorithms which are introduced in class.

6. **(10 points)** Please give the fixed-size, fixed-time, and memory-bounded speedup formula for the general PPT (non-pivoting) tridiagonal solver where the communication overhead is considered. Here general means that the algorithm, so the formula, is general for any given number of processors and for any n by n matrix.

7. **(10 points)** Please give the fixed-size, fixed-time, and memory-bounded speedup formula for the general PDD tridiagonal solver where the communication overhead is considered. Here general means that the algorithm, so the formula, is general for any given number of processors and for any n by n matrix.

8. **(10 points)** In the parallel formulations of bitonic sort, we assumed that we had p processors available to sort n items. Show how the algorithm needs to be modified when only p/2 processors are available.

9. **(10 points**) Show that, in the hypercube formulation of bitonic sort, each bitonic merge of sequences of size 2k is performed on a k -dimensional hypercube and each sequence is assigned to a separate hypercube.

10. **(10 points)** Excise 9.17, Textbook "Introduction of Parallel Computing" (2nd Edition)

11. **(15 points)** In the algorithm shown, assume a decomposition such that each execution of Line 7 is a task. Draw a task-dependency graph and a task-interaction graph.

```
1.  procedure FFT_like_pattern(A, n)
2.  begin
3.      m := log2 n;
4.      for j := 0 to m - 1 do
5.          k := 2^j;
6.          for i := 0 to n - 1 do
7.              A[i] := A[i] + A[i XOR 2^j];
8.      endfor
9.  end FFT_like_pattern
```

12. **(5 points)** In the above algorithm, if n = 16, devise a good mapping for:
    a) 16 processes
    b) 8 processes

13. **(10 points)** Consider seven tasks with running times of 1, 2, 3, 4, 5, 5, and 10 units, respectively. Assuming assign a work to a process does not take any time, compute the best- and worst-case speedup for a centralized scheme for dynamic mapping with two processes.

Note: We encourage collaboration between you and your classmates. Discuss various approaches and techniques to better understand the questions. However, we do NOT allow copying solutions or code. This is considered as cheating and falls under IIT code of honor. Penalties will be enforced. Please make sure you write your own solutions.
GOOD LUCK!