Homework 1 – Parallel Computing, Architectures, and Performance Theory

Due Date: 4/21 @ 5pm

Parallel Computing

1. What is a parallel computer? Can a computer with only one processing core be considered a parallel computer? Explain.
2. What are the three components of parallelism?
3. When executing the Linpack benchmark, is it possible for a parallel computer A to have a larger Rmax than a parallel computer B, but a smaller Nmax?
4. What are some reasons why one would utilize parallel computing? Why is it important to study parallel computing today more than any other time before?

Parallel Architectures

1. What is the difference between a shared memory parallel system and a distributed memory parallel system? Give two advantages for each.
2. What is a NUMA parallel architecture and why was it invented?
3. Why do we have multi-core processors? Are not single-core processors good enough? Why is multi-core a disruptive technology from the point of view of parallel computing? (The high-performance computing (HPC) community oftern refers to a new technology as ”disruptive” if it has the potential of causing a change in how HPC will be done going forward, different from the status quo, often with the benefit of achieving much better performance.)
4. For large-scale parallel systems, the interconnection network is key. Would you agree? Explain.

Parallel Performance Theory

1. What distinguishes “Amdahl’s Law” from “Gustafson-Baris’ Law” in respects to parallel speedup?
2. For a given problem size, why does the efficiency go down as the number of processing elements increase? Is this always true?
3. Suppose you are comparing 2 algorithms, A and B, for the same problem. Suppose that algorithm A has better strong scaling than algorithm B on a parallel machine.

(a) Will algorithm A always have better weak scaling than algorithm B on this machine? Explain.

(b) Is it possible that algorithm B will have better strong scaling than algorithm A on a different parallel machine?

4. Amdahl's Law is defined with respect to the fastest sequential execution time. Suppose the fastest sequential algorithm on machine A, S_A, is not the fastest sequential algorithm on machine B, S_B. Does this cause a problem when comparing speedups between the two machines? Explain.

5. Consider the problem of computing the dot product of two vectors, A and B, each of length N. The dot product is defined as:

$$A \cdot B = \sum_{i=1}^{n} A_i * B_i$$

(a) Describe how you would parallelize this problem.

(b) Assume that multiplying two numbers takes 4 units of time, adding two numbers takes 2 units of time, and communicating one number between two processing elements takes 50 units of time. What is the parallel runtime, speedup, and efficiency of your parallel algorithm when run on P processing elements? You can assume N and P are a power of 2. If you can, write your answer in terms of computation time and communication time components.

(c) Calculate the speedup and efficiency assuming that the problem for P=1 is that of computing the dot product for two vectors of length 256. Use P=1, 4, 16, 64, and 256, and assume the same time costs as in b.

(d) When executing on 64 processors, how large would N have to be to achieve the same efficiency as achieved on 4 processors for N=256?