CS 211: High Performance Computing Project 4

Performance Analysis of Parallel Programs

Due at 11:59 PM on December 3, 2018

1. Benchmarking of a sequential program reveals that 95 percent of the execution time is spent inside functions that are amenable to parallelization. What is the maximum speedup we could expect from executing a parallel version of this program on 10 processors?

2. For a problem size of interest, 6 percent of the operations of a parallel program are inside I/O functions that are executed on a single processor. What is the minimum number of processors needed in order for the parallel program to exhibit a speedup of 10?

3. What is the maximum fraction of the computation that are inherently sequential if a parallel application is to achieve a speedup of 50 over its sequential counterpart?

4. Shauna's parallel program achieves a speedup of 9 on 10 processors. What is the maximum fraction of the computation that may consist of inherently sequential operations?

5. Brandon's parallel program executes in 242 seconds on 16 processors. Through benchmarking he determines that 9 seconds is spent performing initializations and cleanup on one processor. During the remaining 233 seconds, all 16 processors are active. What is the scaled speedup achieved by Brandon's program?

6. Courtney benchmarks one of her parallel programs executing on 40 processors. She discovers that it spends 99 percent of its time inside parallel code. What is the scaled speedup of her program?

7. If a parallel program achieves a speedup of 9 with 10 processors, is it possible to achieve a speedup of 90 with 100 processors if it is ran with the same problem size on the same parallel platform? Why?

8. Assume a parallel program takes 1000 seconds to finish when using 1 processor and 500 seconds to finish when using 4 processors. What is the minimum time to finish the program when using 16 processors? Assume the problem size is fixed.

9. Let $n \geq f(p)$ denote the isoefficiency relation of a parallel system and M(n) denote the amount of memory required to store a problem of size n. Use the scalability function to rank the parallel systems shown below from most scalable to least scalable.

   a. f(p) = Cp and M(n) = n^2
   b. f(p) = $C\sqrt{p}\log p$ and M(n) = n^2
   c. f(p) = $C\sqrt{p}$ and M(n) = n^2
   d. f(p) = $Cp\log p$ and M(n) = n^2
   e. f(p) = Cp and M(n) = n
   f. f(p) = p^c and M(n) = n. Assume 1 < C < 2.
   g. f(p) = p^c and M(n) = n. Assume C > 2.

10. Assume the computation time for your sequential matrix-matrix multiplication program is $2n^3$, and, in the parallel version of the program, the communication time is $16n^2\log_2 p$. For a problem size n, the total memory needed for the algorithm is $24n^2$ bytes. If your parallel computer has 1024 cores with 1 Gbytes DRAM per core, what is the maximum speed up you can achieve on your computer? If you want to achieve a speed up of 256, what is the minimum problem size you need to run your program with?