Assignment 3
Due: 11:59 PM, March 6

Solve Problem 2.10, 2.14, 2.15, 2.19, and 2.20 in the textbook (1st edition).

2.10. Suppose a program must execute 10^{12} instructions in order to solve a particular problem. Suppose further that a single processor system can solve the problem in 10^6 seconds (about 11.6 days). So, on average, the single processor system executes 10^6 or a million instructions per second. Now suppose that the program has been parallelized for execution on a distributed-memory system. Suppose also that if the parallel program uses p processors, each processor will execute $10^{12}/p$ instructions and each processor must send $10^9(p-1)$ messages. Finally, suppose that there is no additional overhead in executing the parallel program. That is, the program will complete after each processor has executed all of its instructions and sent all of its messages, and there won't be any delays due to things such as waiting for messages.

- **a.** Suppose it takes 10^{-9} seconds to send a message. How long will it take the program to run with 1000 processors, if each processor is as fast as the single processor on which the serial program was run?
- **b.** Suppose it takes 10^{-3} seconds to send a message. How long will it take the program to run with 1000 processors?

2.14. To define the bisection width for indirect networks, the processors are partitioned into two groups so that each group has half the processors. Then, links are removed from *anywhere* in the network so that the two groups are no longer connected. The minimum number of links removed is the bisection width. When we count links, if the diagram uses unidirectional links, two unidirectional links count as one link. Show that an eight-by-eight crossbar has a bisection width less than or equal to eight. Also show that an omega network with eight processors has a bisection width less than or equal to four.

2.15. **a.** Suppose a shared-memory system uses snooping cache coherence and write-back caches. Also suppose that core 0 has the variable `x` in its cache, and it executes the assignment `x = 5`. Finally suppose that core 1 doesn't have `x` in its cache, and after core 0's update to `x`, core 1 tries to execute `y = x`. What value will be assigned to `y`? Why?

- **b.** Suppose that the shared-memory system in the previous part uses a directory-based protocol. What value will be assigned to `y`? Why?
- **c.** Can you suggest how any problems you found in the first two parts might be solved?

2.19. Suppose $T_{\text{serial}} = n$ and $T_{\text{parallel}} = n/p + \log_2(p)$, where times are in microseconds. If we increase p by a factor of k, find a formula for how much we'll need to increase n in order to maintain constant efficiency. How much should we increase n by if we double the number of processes from 8 to 16? Is the parallel program scalable?

2.20. Is a program that obtains linear speedup strongly scalable? Explain your answer.