CSC/ECE 506: Architecture of Parallel Computers
Problem Set 4
Due Thursday, April 29, 2004

Problems 1, 2, 3 and 4 will be graded. There are 70 points on these problems. Note: You must do all the problems, even the non-graded ones. If you do not do some of them, half as many points as they are worth will be subtracted from your score on the graded problems.

Problem 1. CS&G 8.9 (10 points) When a variable exhibits migratory sharing, a processor that reads the variable will be the next one to write it. What kinds of protocol optimizations could you use to reduce traffic and latency in this case, and how would you detect the situation dynamically? Describe a scheme or two in some detail. (Hint: Think about different ways to give a requesting processor exclusive access to a block, even when only shared access is requested when it is likely that the data exhibits this migratory sharing pattern.)

Problem 2. CS&G 9.20 (20 points) Two processors $P_1$ and $P_2$ are executing the following code fragments under the sequential consistency (SC) and release consistency (RC) models.

```
\begin{align*} 
P_1 & \quad P_2 \\
\text{LOCK (L1)} & \quad \text{LOCK(L1)} \\
A = 1 & \quad x = A \\
B = 2 & \quad y = B \\
\text{UNLOCK(L1)} & \quad x1 = A \\
\text{UNLOCK(L1)} & \quad x2 = B \\
\end{align*}
```

Assume an architecture where both read and write misses take 100 cycles to complete. However, you can assume that accesses that are allowed to be overlapped under the consistency model are indeed fully overlapped. Acquiring a free lock from another processor or unlocking a lock takes 100 cycles, and no overlap is possible with lock-unlock operations from the same processor. Assume all the variables and locks are initially uncached and all locks are unlocked, that all memory locations are initialized to 0, and that all memory locations are distinct and map to different indices in the caches (i.e. different cache lines)

(a) What are the possible outcomes for $x$ and $y$ under SC? Under RC?

(b) Assume $P_1$ gets the lock first. After how much time from the start of $P_1$’s lock operation will $P_2$ complete all its operations while satisfying the sufficient conditions for SC described in Chapter 5? What if it satisfies the sufficient conditions for RC described in this chapter?

Problem 3. CS&G 9.33 (20 points) Trace the path of a write reference in (a) pure CC-NUMA, (b) a flat COMA, (c) an SVM with automatic update, (d) an SVM protocol without an automatic update, and (e) simple COMA. (Hint: See how it was done for read references in Chapter 9).
Problem 4. (20 points) (a) Show that an $8 \times 8$ matrix can be transposed in three perfect shuffles when it is stored as a vector of length 64. Hint: Let $A = [a_{ij}]$ be the original matrix and $B = [b_{ij}]$ be its transpose. By calculations on the subscripts, show that after three transpositions, the element originally at position $ij$ is now at position $ji$.

(b) Prove that $m$ shuffles transpose a matrix of size $2^m \times 2^m$.

Problem 5. (30 points) Below are diagrams of the eight-element cube and omega interconnection networks.

(a) Suppose that we desire to route a data item from processor $i$ to processor $(i+k) \mod 8$. Each of the cells can be set in two ways: 0 denotes pass-through, while 1 denotes interchange. (The cells in the omega network can also be set to broadcast, but this capability is not used in this problem.) To perform this routing in the cube network, how should we set each of the cells that the data passes through? Give an algorithm for setting the cell in each stage $j$, where $j = 0, 1, 2$. Hint: Consider the binary representation of $i$ and $(i+k)$.

(b) Notice that the effect of a shuffle is to send data from processor $(d_2 d_1 d_0)\text{2}$ to processor $(d_1 d_0 d_2)\text{2}$ (i.e., to rotate the bits of the processor number left by 1). To route data from processor $i$ to processor $(i+k) \mod 8$ in the omega network, how should we set each of the cells that the data passes through? Give an algorithm for setting the cell in each stage $j$, where $j = 0, 1, 2$.

(c) The $n$-cube and omega networks are functionally equivalent by some relabeling techniques provided one considers only one-to-one connections. The figure below illustrates this technique.

The shuffle-exchange network for $n = 8$. The same network with switch boxes repositioned.

Draw a diagram that illustrates this technique for a 16-element shuffle-exchange network.