Multiprocessor Networks, CS294-5 Handout 3
Prof. Eric A. Brewer October 31, 1994

Problem Set 1

This problem set is worth 10% of the overall grade; it is due Wednesday November 9th.The primary goal is to investigate some of the readings with more depth than was possible in lecture. Discussion of problems with other students is allowed, but you must write up your answers independently. Legibility matters only to the extent that I can read your submission.

1. The CM-5 allows time sharing as well as space sharing, and uses gang scheduling to ensure that all nodes within a partition execute the same job. During a context switch, what happens to the messages in the network? Do they affect the next user? Are users penalized for having messages in the network when their time slice expires?

2. What are the primitive operators for the CM-5 control network? How do you implement unsigned minimum (the minimum of unsigned 32-bit words) on top of this set? Give an algorithm for the median of *n* values (one from each node) that uses only the control network for communication (hard). The CM-5 libraries build 32-bit floating-point addition on top of the control network; how is this done (roughly)? Hint and follow-up question: why doesn't this work for double-precision floating point? How could you implement 64-bit floating-point scan operations?

3. The normal-case bidirectional transfer rate for Strata on the CM-5 is one active message in each direction per 99 cycles, or 10.66 MB/sec. (16 bytes of payload $\Rightarrow$ $2\frac{16\,(33\text{ Mhz})}{99} = 10.66$ MB/sec). Assume that floating-point conversion due to heterogeneous nodes requires 50 cycles per 32-bit floating-point value. What is the net bandwidth counting the conversion? What if the conversion is done on the sender side? Where should the conversion be done and why?

4. Give some specific evidence that reducing the average packet size leads to lower latency and higher network utilization (ignoring software overhead). Next, give some specific evidence that smaller messages lead to lower throughput due to software overhead. Given these two conflicting items, what is required to achieve maximum network performance?

5. Clark and Tennenhouse argue that only the data-manipulation phases of TCP really affect performance. Kay and Pasquale argue that most of the overhead comes from non-data operations. When is data manipulation the dominant source of overhead? Answers should be complete and reasonably precise, which should take about half of a page.

6. What happens in METRO if the receiver is not expecting a block transfer? What should the network interface do to reduce this problem? What happens when there are multiple senders for one receiver? Classify the techniques from Brewer and Kuszmaul in terms of their usefulness for METRO.

7. Assume that we have a strided block transfer that can send a sequence of evenly spaced, identically sized blocks with a single coprocessor call, so that the transfer occurs concurrently with the main computation. Assume we are solving a 3D PDE problem iteratively, which means that after each iteration the six *faces* of our local 3D volume must be transferred to the corresponding six nearest neighbors. The local volume is a 100x100x100 array of doubles. How many calls are required to send all of the data from one node? How many messages would be required using only contiguous block transfers? (You're not allowed to send any extra data; for example, sending the entire volume to all neighbors would require only six messages.) If we assume that it takes $100 + 0.25k$ cycles for any k-byte transfer, how much faster is the strided transfer per iteration?

8. List several reasons why adaptive routing may be a bad idea. When does it make sense to use adaptive routing? There is no perfect answer here; just give a reasonable argument that covers several variables including (at least) topology, traffic pattern, fault tolerance, and end-to-end software overhead.