

**CSE/ESE 569M**  
**Parallel Architectures and Algorithms**  
**Assignment #1**

Due: January 29, 2009.

1. Consider a simple 2D finite difference problem where at each step every point in a matrix is updated by a weighted average of its four neighbors:

$$A[i,j] = A[i,j] - w(A[i-1,j]+A[i+1,j]+A[i,j-1]+A[i,j+1])$$

All the values are 64-bit floating-point numbers. Assuming one element per processor and  $1,024 \times 1,024$  elements, how much data must be communicated per step? Explain how this computation could be mapped onto 64 processors so as to minimize the data traffic. Compute how much data must be communicated per step.

2. To move an  $n$ -byte message along  $H$  links in an unloaded store-and-forward network takes time  $H(n/W) + (H-1)R$ , where  $W$  is the raw link bandwidth and  $R$  is the routing delay per hop. In a network with cut-through routing, this takes time  $n/W + (H-1)R$ . Consider an  $8 \times 8$  grid consisting of 400 MB/s links and routers with 25 ns of delay. What is the minimum and maximum time to move a 64 byte message through the network? A 256 byte message?
3. Suppose we have a machine with a 64 bit wide bus running at 800 MHz. It takes 2 bus cycles to arbitrate for the bus and present the address. The cache line size is 32 bytes and the memory access time is 100 ns. What is the latency for a read miss? What bandwidth is obtained on this transfer?
4. Suppose this 32 byte line is transferred to another processor and the communication architecture imposes a startup cost of  $0.2 \mu\text{s}$  and a data transfer bandwidth of 200 MB/s. What is the total latency of the remote operation?
5. The serial code that implements a Monte Carlo simulation to estimate the value of  $\pi$  is available on the class web page. In addition, template code that implements a master-slave parallel computation using the shared memory programming paradigm is available on the class web page. Use these to design, implement, and test a parallel implementation of the serial code. Report on the measured performance of your implementation as a function of the number of threads created (ranging from 1 to 4) and the number of iterations executed in the simulation (ranging enough to see the interesting effects that occur).

Please combine all of your results above (including the source files for problem 5 and any makefiles and/or scripts needed to build and execute the code, with usage instructions) into one .zip file and upload it via CMS. The text results can be in any of the following forms: .txt, .doc, .docx, .rtf, or .pdf.