Homework 6 (due April 10, 2014)

Questions 1: Consider the "invalidate" snooping protocol for "write-back" caches in a bus-based system with three processors P1, P2, and P3, each with a direct-mapped cache and assume that the size of a cache line (block) is two words. The following sequence of memory operations access two blocks, A and B, mapped to the same location in the cache. Variables x and y are stores in block A and variable w and z are stored in block B.

P_1 writes x = 4
P_2 writes z = 8
P_3 reads y
P_1 reads y
P_3 writes x = 10
P_1 reads w
P_2 reads z
P_1 writes z = 20

Assume that initially, x = y = w = z = 3 and that the cache is initially empty. For each memory access, determine

- The content and state of the cache line in each processor
- The bus operations caused by the cache coherence protocol

Questions 2: Consider the parallel sum algorithm shown on page 30 of the class slides. Plot in a graph the speedup that is expected when the algorithm is executed on 16 processors for N = 200, 400, 600, 800, 1000, … , 2000.

Questions 3: Compile and execute the "compute pi" program (in file pi.c -- see notes below) with 50000 sample points and P =1, 2, 4 and 8 threads on "unixs1.cis.pitt.edu". Report the execution times in a table and draw the speed up curve. Repeat the experiment with 5000 and 500000 points and plot the three speedup curves on the same graph. Comment on your results.

Questions 4: Rewrite the matrix/vector multiplication program, mat_vec.c so that it accepts a second command line argument, P. Your program should then execute on P threads with each thread computing N/P elements of the product vector, where N is the dimension of the product vector. Find the execution time and speed up on unixs1.cis.pitt.edu with P = 1, 2, 4 and 8 and N = 100, 500 and 1000. Comment on your results.
NOTE: in addition to including the results in your homework, you should email your source file(s) to the TA before class time on the 25^{th}.

Notes:

1) You can login to unixs1.cis.pitt.edu using "ssh" and your CSSD login name and password (example "ssh unixs1.cis.pitt.edu"). It is a quad-core machine (with an "afs" file system).
2) Once on unixs1 you can copy pi.c and mat_vec.c using

```
cp  /afs/cs.pitt.edu/public/courses/1541/pi.c   .
cp  /afs/cs.pitt.edu/public/courses/1541/mat_vec.c   .
```

3) To compile a c program (for example pi.c) on "unixs1" use

```
gcc  -D_REENTRANT -lpthread -lm -o pi pi.c
```

4) To execute the compiled program "pi" use "./pi arguments". See the comments in pi.c for the description of the arguments.