CS4961: Parallel Programming Final Take-Home Quiz
December 9, 2010
Due December 16, 2010

Instructions:
This is a take-home open-book, open-note exam. The format is that of a long
homework. The exam is due by 5PM, Thursday, December 16. Use the CADE handin
program, as follows:
    handin cs4961 final <file>
or drop off in my office no later than December 16.

The goal of the exam is to reinforce your understanding of issues we have studied in
class.
CS4961: Parallel Programming
Final Take-Home Quiz
December 9, 2010

I. Definitions (10 points)
Provide a very brief definition of the following terms:

a. Adjacency matrix
b. Map in Map/reduce
c. Divergent branches
d. Partitioned global address space languages
e. Collective communication

II. Short answer (20 points) [Fill in the blanks]
a. In CUDA, we can synchronize a collection of threads using a ____________ construct. This type of synchronization, where a thread waits to proceed until all threads have reached this point, is called a ________________.

b. List two properties of algorithms on sparse graphs that make efficient parallel implementations difficult.

c. A __________ operation copies local data into a remote processor's memory without notifying the remote processor. This type of “communication” is called ________________.

d. __________ languages permit programs whose correct execution depends on the number of processors, while __________ languages result in programs that always produce the same output on the same input regardless of number or arrangement of processors.
III. Problem Solving (60 points)
In this set of six questions, you will be asked to provide code solutions to solve particular problems. This portion of the exam may take too much time if you write out the solution in detail. I will accept responses that sketch the solution, without necessarily writing out the code or worrying about correct syntax. Just be sure you have conveyed the intent and issues you are addressing in your solution.

a. Sketch an MPI implementation of the Red/Blue computation (defined by Problem #10 on page 111 of the textbook) using non-blocking communication.

b. (i) The following 5x5 matrix is represented in a dense form. Convert it to a CSR representation (compressed sparse row) format.

\[
A = \begin{bmatrix}
4 & 0 & 0 & 0 & 0 \\
0 & 3 & 0 & 1 & -1 \\
2 & 5 & 0 & 0 & 9 \\
0 & 0 & 6 & 0 & 0 \\
0 & 0 & 0 & -2 & 0
\end{bmatrix}
\]

(ii) Provide a dense matrix representation of the following undirected graph.

(iii) Now provide a sparse representation of the graph, either as a sparse matrix as in (i), or an adjacency list.
c. Given the following sequential code, sketch out two CUDA implementations for just the kernel computation (not the host code). Determine the memory access patterns and whether you will need synchronization. Your two versions should (a) use only global memory; and, (b) use both global and shared memory. Keep in mind capacity limits for shared memory and limits on the number of threads per block.

... 
int a[1024][1024], b[1024];

for (i=0; i<1020; i++) { 
    for (j=0; j<1024-i; j++) { 
        b[i+j] += a[j][i] + a[j][i+1] + a[j][i+2] + a[j][i+3]; 
    }
}
}
III. Essay Question (10 points)
Select ONE of the following programming models we studied at the end of the semester: Map/reduce, transactional memory, Chapel and global view languages, CUDA or MPI. Write a very brief essay describing what types of applications would be most appropriate for that programming model. Also, what are the advantages and disadvantages of using that programming model in terms of both expressibility and performance.