Parallel Algorithms (15-499), Spring 09
Assignment #6 (course project) **Due: April 7 (proposal), April 30 (project)**

The final project of the course will involve developing some algorithm or application to run on a multicore machine. We expect everyone to use either Cilk or OpenMP. You can run your code on Multi6. However, if you really want to use some other language and platform please discuss it with the instructor.

You can work in groups of 1, 2 or 3.

Here is a list some possible ideas for projects, but please feel free to come up with your own ideas.

1. An algorithm for finding all closest pairs for a set of points in two (or three) dimensions.
2. An algorithm for finding all intersecting line segments among a set of 2-dimensional segments.
3. An algorithm for finding the minimum-spanning-tree of a graph.
4. An algorithm for finding the maximum flow in the graph.
5. Some game playing program.
6. An algorithm for some rendering task in graphics. This is only suggested if you have taken the graphics course have have a particular task from that course in mind.
7. Some algorithm from computational biology or other field. Again this is only suggested if you have taken a course on the topic.

The project proposal is due on April 7th. This should be about 1/2 page and describe the problem you plan to solve and the programming language you plan to use. Every group only needs to hand in one proposal and final project. All names should be on the documents.

Your project will be graded based on quality of the code, innovation of the parallel algorithm used, and speedups achieved. A heavier weight will be placed on the first two criteria. If you simply take existing code and put some parallel for loops into it, it would not be considered innovative. Furthermore it is important that you try to make it as parallel as possible even if you will only be running it on 16 threads.

You will be required to hand in a working version of your code that we can compile and run. You will also be required to hand in a 3-5 page writeup of what the code does, how you implemented the algorithm, and some analysis of the parallelism of the algorithm (e.g. how far will it scale). This will be due on the last day of class (April 30th).