CS 5220: Review of Group 20 Homework 1

Group 19: Robert Chiodi (rmc298), Zhiyun Ren (zr54), Sania Nagpal (sn579)

September 21, 2015

What was done well

A lot of effort was clearly put into testing different methods of optimization. The group tried blocking, rearrangement of loop orders, copy optimization, and selectively avoiding copy optimization when the remaining section of the matrix was not an even multiple of the block size.

Blocking was used in order to reduce the working set and fit portions of the matrices into the L2 cache. The allowable size of the block to fit into L2 cache seemed correctly chosen. The smaller than maximum size being optimal is logical since there is some overhead associated with the program.

Changing the index order should give a noticeable result through making vectorization easier for the compiler to perform (by reducing strides in memory). Using more aggressive compiler optimization should provide a boost from correct index order.

Other optimizations were used well to good effect. The overall result is similar to ours before we applied compiler optimizations. These are a great help and should be used.

Possible Improvements

A large and easy to implement improvement will come from using compiler optimizations. These can be found quite readily on the internet. It is not mentioned which compiler is used, however using Intel compilers on Totient should also provide a performance boost over GCC compilers. The effect of a lot of the optimizations already made will be magnified by the use of a well directed compiler. A last suggestion would be allowing the creation of rectangular instead of square blocks, which allows for blocks to be longer in dimensions which will experience unit strides in memory (assuming column major matrix format is kept).