CS 5220: Review of Group 21 Homework 1

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

September 21, 2015

What was done well

Many optimization techniques were tried and the plots comparing results during each stage are very helpful. By comparing results in this manner, it is possible to select the best option at each stage of optimization. The one thing to be weary of, however, is if one optimization decision impacts the other. For instance, if a non-ideal block size when comparing a simple blocked routine is more easily unrolled or vectorized by the compiler, which could lead to a net performance gain.

The manual unrolling of the loop was a good idea and may possibly be combined with knowledge of Totient's architecture to increase vectorization. It may be best to also consult the size of the L1 cache and registers when selecting the size of the unrolled loop.

Possible Improvements

We can not seem to locate one single function with all optimizations implemented, so we are unsure of the final performance of your code, or which optimizations you have kept and tuned.

In our experience, optimization flags can lead to the easiest and quickest optimization. There are other optimization flags not used that could lead to significant improvements. Viewing the ability of the compiler to unroll certain loops can also help improve performance of the code, and is possible using compiler flags.

We are not sure if you have tried combining these optimizations, however it will most likely change the performance of some of the optimization methods due to coupling of the methods. It is hard to compare your performance to our performance since there is not a function with all optimizations implemented. As reference, our performance was mostly constant across all dimension sizes between 7 and 10 GFlops/s.