CS6001 HPC: Assignment 1

1 Theoretical analysis on matrix-matrix multiplication

1. Assume your computer is able to complete 4 double floating-point operations per cycle when operands are in registers and it takes an additional delay of 100 cycles to access any operands that are not in registers. The clock frequency of your computer is 2 Ghz. How long it will take for your computer to finish the algorithm dgemm and dgemm_v respectively for n= 1000? How much time is wasted on accessing operands that are not in registers?

2. Suppose your data cache has 60 lines and each line can hold 10 doubles. You are performing a matrix- matrix multiplication (C=C+A*B) with square matrices of size 10000X10000 and 10X10 respectively. Assume data caches are only used to cache matrix elements which are doubles. The cache replacement rule is least recently used first. Assume no registers can be used to cache intermediate computing results. One-dimensional arrays are used to represent matrices with the row major order.
1) When matrix-matrix multiplication is performed using the simple triple-loop algorithm with single register reuse, there are 6 versions of the algorithm (ijk, ikj, jik, jki, kij, kji). Calculate the number of read cache misses for each element in each matrix for each version of the algorithm when the sizes of the matrices are 10000X10000 and 10X10 respectively. What is the percentage of read cache miss for each algorithm?
2) If matrices are partitioned into block matrices with each block being a 10 by 10 matrix, then the matrix-matrix multiplication can be performed using one of the 6 blocked version algorithms (ijk, ikj, jik, jki, kij, kji). Assume the multiplication of two blocks in the inner three loops uses the same loop order as the three outer loops in the blocked version algorithms. Calculate the number of read cache misses for each element in each matrix for each version of the blocked algorithm when the size of the matrices is 10000. What is the percentage of read cache miss for each algorithm?

2 Practice on matrix-matrix multiplication

1. Test dgemm, dgemm_v, and dgemm with different loop ordering on the Foundry with n= 64, 128, 256, 512, 1024. Measure the time spend in the triple loop for each algorithm. Calculate the performance (in Gflops) of each algorithm. Performance is often defined as the number of floating-point operations performed per second. A performance of 1 GFLOPS means 1 billion of floating-point operations per second. You must use the system default compiler to compile your program. Your test matrices have to be 64-bit double floating point random numbers. Report the maximum difference of all matrix elements between the results of each algorithm and that of dgemm. This maximum difference can be used as a way to check the correctness of your implementation.

2. Assume you have 16 registers to use, please maximize the register reuse in your code and compare your performance with dgemm and dgemm_v.

3. Implement the algorithms in blocked dgemm for loop ordering ijk. Report your execution time on the Foundry. Adjust the block size from 8 to other numbers to see what the optimal block size is. Compare and analyze the performance of your codes for n=1024. Please always verify the correctness of your code.

4. (Optional) Improve your implementation by using both cache blocking and register blocking at the same time. Optimize your block sizes to achieve the best performance for n=1024.