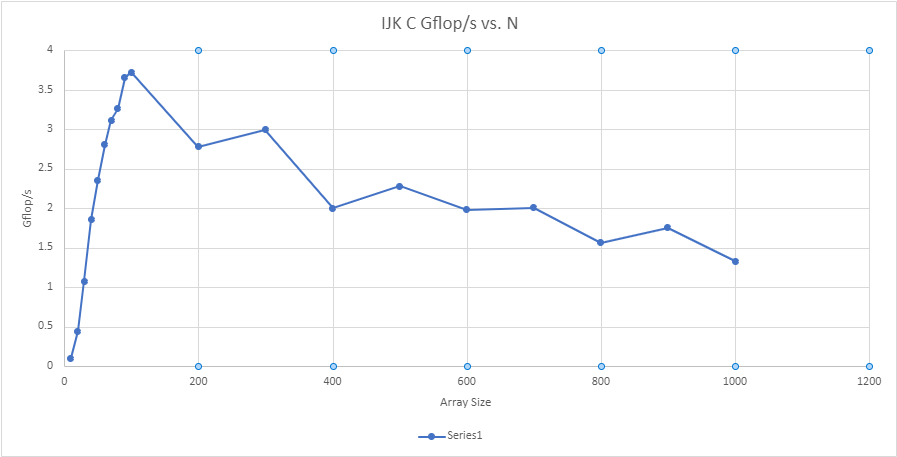
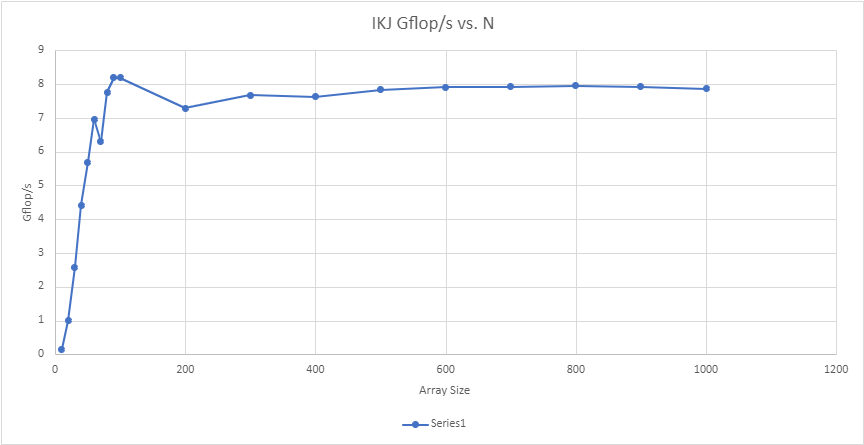
Michael Thorman

CSCI490/Fall 2021

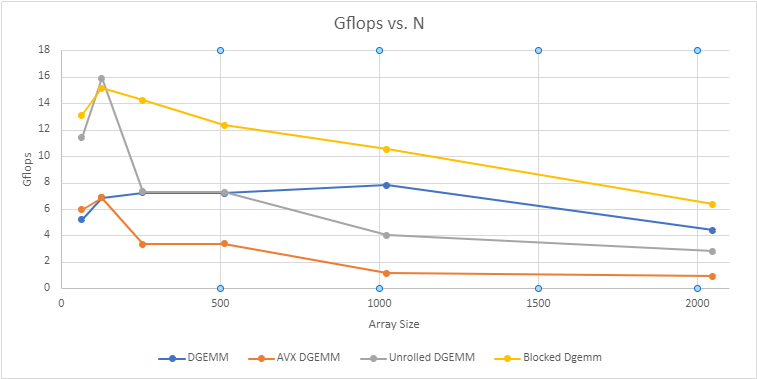
Lab 3

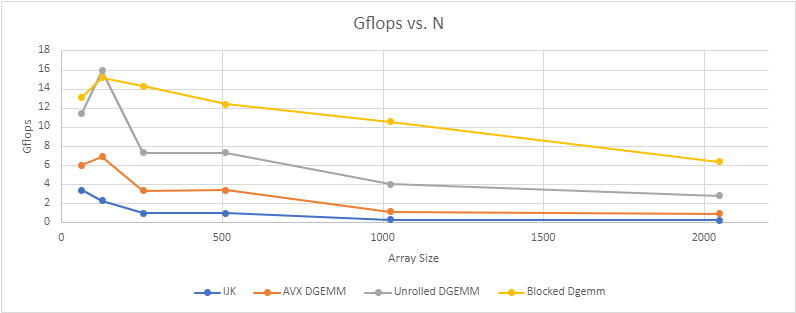
The purpose of this lab is to apply AVX instructions, loop unrolling and software tiling or blocking to improve performance of the DGEMM from Lab 2. In Lab 2, it was found that the IKJ and KIJ versions of naïve matrix multiplication outperformed the other four versions because they were able to reach and sustain higher Gflop/s, or higher computation times, closer to the theoretical peak performance of Big Red 3 (around 40x faster when comparing cycles per inner loop iteration to array size N the larger N got). Furthermore, IKJ and KIJ were able to complete more instructions per cycle and had the lowest data cache miss rate.





After applying AVX instructions, loop unrolling (unroll =4), and software tiling (blocksize=32), I noticed a significant increase in computation time for IJK. With N=128, IJK’s computation time was 15.89 Gflop/s, compared to 3.72 Gflop/s in Lab 2 with N= 100, which was the highest computation time for IJK with N from 10-1000. This large increase in computation time was thanks in most part to loop unrolling, but with software tiling, IJK was able to get its second highest computation time of 15.14 Gflop/s. A computation time of 15.89 Gflop/s puts IJK within 2.16% of Big Red 3’s theoretical peak performance (this is compared to 0.51% from Lab 2.





These graphs also show that the fully optimized version of IKJ performed better than the IKJ version. The highest Gflop/s IKJ was able to reach with N between 10 and 1000 was 8.2 Gflop/s.

As you can see from the following table, N should be a multiple of block size and and block size should be a multiple of unroll\*4. Moreover, you can increase the computation speed for the unrolled and blocked versions of IJK by changing the values. You will also notice that the computation speed does not change for the normal IJK algorithm or the optimized AVX version. The data I collected for the graphs was using blocksize = 32 and unroll = 4, and if the value for unroll is changed to 1 you will notice a slightly higher wall clock time for N=64, but as unroll gets closer to 4, then wall clock time gets closer to value recorded for N=64 unroll =4. Similarly, as blocksize gets closer to 32, the wall clock time gets closer to the wall clock time recorded for N=64 blocksize =32.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N=64 | Unroll | Blocksize | Wall clock time | Gflop/s compared to Blocksize = 32 and Unroll = 4 |
| IJK | 1 | 1 | 161 | normal |
| AVX | 1 | 1 | 85 | normal |
| UNROLL | 1 | 1 | 79 | lower |
| BLOCK | 1 | 1 | 387 | lower |
| IJK | 1 | 2 | 156 | normal |
| AVX | 1 | 2 | 83 | Normal |
| UNROLL | 1 | 2 | 80 | lower |
| BLOCK | 1 | 2 | 224 | Lower |
| IJK | 1 | 4 | 160 | Normal |
| AVX | 1 | 4 | 83 | Normal |
| UNROLL | 1 | 4 | 83 | Lower |
| BLOCK | 1 | 4 | 54 | Normal |
| IJK | 1 | 8 | 155 | normal |
| AVX | 1 | 8 | 84 | Normal |
| UNROLL | 1 | 8 | 83 | lower |
| BLOCK | 1 | 8 | 64 | Normal |
| IJK | 2 | 2 | 155 | normal |
| AVX | 2 | 2 | 82 | Normal |
| UNROLL | 2 | 2 | 50 | normal |
| BLOCK | 2 | 2 | 249 | Lower significantly |
| IJK | 2 | 4 | 161 | normal |
| AVX | 2 | 4 | 86 | normal |
| UNROLL | 2 | 4 | 55 | Normal |
| BLOCK | 2 | 4 | 71 | Lower, but going up |
| IJK | 2 | 8 | 161 | Normal |
| AVX | 2 | 8 | 82 | Normal |
| UNROLL | 2 | 8 | 50 | Normal |
| BLOCK | 2 | 8 | 58 | normal |
| N=128 |  |  |  |  |
| UNROLL | 3 | 1 | 338 | lower |
| BLOCK | 3 | 1 | 11074 | lower |
| UNROLL | 3 | 2 | 347 | lower |
| BLOCK | 3 | 2 | 3185 | lower |
| UNROLL | 3 | 4 | 356 | lower |
| BLOCK | 3 | 4 | 865 | lower |
| UNROLL | 3 | 8 | 348 | lower |
| BLOCK | 3 | 8 | 539 | lower |
| UNROLL | 3 | 16 | 339 | same |
| BLOCK | 3 | 16 | 545 | same |
| UNROLL | 3 | 32 | 338 | same |
| BLOCK | 3 | 32 | 377 | lower |
| UNROLL | 4 | 1 | 270 | same |
| BLOCK | 4 | 1 | 12242 | lower |
| UNROLL | 4 | 8 | 264 | same |
| BLOCK | 4 | 8 | 600 | lower |
| UNROLL | 4 | 16 | 266 | same |
| BLOCK | 4 | 16 | 283 | same |
| UNROLL | 4 | 32 | 265 | same |
| BLOCK | 4 | 32 | 259 | same |
| UNROLL | 4 | 64 | 275 | same |
| BLOCK | 4 | 64 | 281 | same |
| UNROLL | 5 | 64 | 314 | lower |
| BLOCK | 5 | 64 | 380 | lower |
| UNROLL | 6 | 64 | 310 | lower |
| BLOCK | 6 | 64 | 296 | lower |
| UNROLL | 6 | 128 | 292 | lower |
| BLOCK | 6 | 128 | 286 | lower |
| UNROLL | 8 | 128 | 257 | same |
| BLOCK | 8 | 128 | 252 | same |
|  | 8 | 256 |  | BUS ERROR |
|  | 8 | 2048 |  | BUS ERROR |
|  | 9 | 256 |  | BUS ERROR |
| UNROLL | 2 | 128 | 355 | Lower |
| BLOCK | 2 | 128 | 470 | lower |
|  | 2 | 256 |  | BUS ERROR |
| UNROLL | 8 | 32 | 262 | same |
| BLOCK | 8 | 32 | 218 | HIGHER |

Every optimized DGEMM completed more instructions per cycle than the original DGEMM. As the number of N increased, the IPC decreased for the original DGEMM. The IPC decreased as well for the optimized versions, but nearly as much. In every case of N, the original DGEMM completed less IPC than the optimized versions. With N=2048, there was an 83.95% improvement using the blocked or tiled optimization over the original. Moreover, every optimized version of DGEMM had less instruction cache misses and misses/instruction. Therefore, optimizing DGEMM made it more efficient and allowed it to access main memory with less latency than normal.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N=2048** | **PAPI\_TOT\_CYC** | **PAPI\_TOT\_INS** | **IPC** | **PAPI\_L1\_DCM** | **PAPI\_L1\_ICM** | **Miss/Inst** |
| **DGEMM\_IJK** | |  |  | | --- | --- | | |  | | --- | | 234576507354 | | | |  |  | | --- | --- | | |  | | --- | | 31335703783 | | | 0.13 | |  | | --- | | 13995877741 | | |  | | --- | | 102962 | | 0.45 |
| **AVX** | |  | | --- | | 59935623856 | | |  | | --- | | 21484283283 | | 0.36 | |  | | --- | | 3497096972 | | |  | | --- | | 5266 | | 0.16 |
| **Unrolled** | |  | | --- | | 19178659047 | | |  | | --- | | 5373693387 | | 0.28 | |  | | --- | | 1412307527 | | |  | | --- | | 10460 | | 0.3 |
| **Blocked** | |  | | --- | | 8464173144 | | |  | | --- | | 6865627118 | | 0.81 | |  | | --- | | 1237718066 | | |  | | --- | | 2157 | | 0.18 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N=1024** | **PAPI\_TOT\_CYC** | **PAPI\_TOT\_INS** | **IPC** | **PAPI\_L1\_DCM** | **PAPI\_L1\_ICM** | **MISS/INST** |
| **DGEMM\_IJK** | |  | | --- | | 25895432955 | | |  | | --- | | 3941619571 | | 0.15 | |  | | --- | | 1083529055 | | |  | | --- | | 2098 | | 0.27 |
| **AVX** | |  | | --- | | 6041019810 | | |  | | --- | | 2686716837 | | 0.44 | |  | | --- | | 303974157 | | |  | | --- | | 497 | | 0.11 |
| **Unrolled** | |  |  | | --- | --- | | |  | | --- | | 1689274468 | | | |  |  | | --- | --- | | |  | | --- | | 672335051 | | | 0.40 | |  | | --- | | 143775768 | | |  | | --- | | 519 | | 0.21 |
| **Blocked** | |  |  | | --- | --- | | |  | | --- | | 643279798 | | | |  | | --- | | 858213177 | | 1.33 | |  | | --- | | 157302786 | | |  | | --- | | 181 | | 0.18 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N=512** | **PAPI\_TOT\_CYC** | **PAPI\_TOT\_INS** | **IPC** | **PAPI\_L1\_DCM** | **PAPI\_L1\_ICM** | **MISS/INST** |
| **DGEMM\_IJK** | |  | | --- | | 867649539 | | |  | | --- | | 498870682 | | 0.57 | |  | | --- | | 134997591 | | |  | | --- | | 278 | | 0.27 |
| **AVX** | |  | | --- | | 249848732 | | |  | | --- | | 336135510 | | 1.34 | |  | | --- | | 37937895 | | |  | | --- | | 100 | | 0.11 |
| **Unrolled** | |  | | --- | | 114947084 | | |  | | --- | | 84198030 | | 0.73 | |  | | --- | | 17982332 | | |  | | --- | | 130 | | 0.21 |
| **Blocked** | |  | | --- | | 643279798 | | |  | | --- | | 858213177 | | 1.33 | |  | | --- | | 19418264 | | |  | | --- | | 82 | | 0.022 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N=256** | **PAPI\_TOT\_CYC** | **PAPI\_TOT\_INS** | **IPC** | **PAPI\_L1\_DCM** | **PAPI\_L1\_ICM** | **MISS/INST** |
| **DGEMM\_IJK** | |  | | --- | | 107694025 | | |  | | --- | | 63903325 | | 0.59 | |  | | --- | | 16953303 | | |  | | --- | | 261 | | 0.27 |
| **AVX** | |  | | --- | | 30945418 | | |  | | --- | | 42091268 | | 1.36 | |  | | --- | | 4753238 | | |  | | --- | | 95 | | 0.11 |
| **Unrolled** | |  | | --- | | 14362302 | | |  | | --- | | 10564005 | | 0.74 | |  | | --- | | 2253166 | | |  | | --- | | 75 | | 0.21 |
| **Blocked** | |  | | --- | | 7358702 | | |  | | --- | | 13410774 | | 1.82 | |  | | --- | | 2402198 | | |  | | --- | | 65 | | 0.18 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N=128** | **PAPI\_TOT\_CYC** | **PAPI\_TOT\_INS** | **IPC** | **PAPI\_L1\_DCM** | **PAPI\_L1\_ICM1** | **MISS/INST** |
| **DGEMM\_IJK** | |  | | --- | | 5534241 | | |  | | --- | | 8375508 | | 1.51 | |  |  | | --- | --- | | |  | | --- | | 2138053 | | | |  | | --- | | 189 | | 0.26 |
| **AVX** | |  | | --- | | 1745969 | | |  | | --- | | 5280227 | | 3.0 | |  | | --- | | 600312 | | |  | | --- | | 84 | | 0.11 |
| **Unrolled** | |  | | --- | | 748667 | | |  | | --- | | 1330484 | | 1.78 | |  |  | | --- | --- | | |  | | --- | | 285206 | | | |  | | --- | | 67 | | 0.21 |
| **Blocked** | |  | | --- | | 780857 | | |  | | --- | | 1676717 | | 2.15 | |  | | --- | | 159200 | | |  | | --- | | 55 | | 0.09 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N=64** | **PAPI\_TOT\_CYC** | **PAPI\_TOT\_INS** | **IPC** | **PAPI\_L1\_DCM** | **PAPI\_L1\_ICM** | **MISS/INST** |
| **DGEMM\_IJK** | |  | | --- | | 400062 | | |  | | --- | | 1144851 | | 2.86 | |  | | --- | | 16110 | | |  |  | | --- | --- | | |  | | --- | | 142 | | | 0.014 |
| **AVX** | |  | | --- | | 210626 | | |  | | --- | | 664914 | | 3.16 | |  | | --- | | 11852 | | |  | | --- | | 84 | | 0.017 |
| **Unrolled** | |  | | --- | | 95480 | | |  | | --- | | 168956 | | 1.77 | |  | | --- | | 8625 | | |  | | --- | | 46 | | 0.05 |
| **Blocked** | |  | | --- | | 80440 | | |  | | --- | | 209825 | | 2.61 | |  | | --- | | 3033 | | |  | | --- | | 62 | | 0.014 |