Accelerating Matrix Multiplication

Namrata Priyadarshani
cs1170353@cse.iitd.ac.in

Musunuru Saurav
cs1170352@cse.iitd.ac.in

I. Linear Algebra Libraries

We tried to accelerate matrix multiplication using different linear algebra libraries. We used Intel MKL[1] and linked it using [2] which provides highly optimized, threaded and vectorized maths function that enhances performance on Intel processor architecture. It consists of Intel MKL DGEMM function which is highly optimized for matrix-multiplication accelerating the matrix-multiplication over 30 times the normal matrix-multiplication. We also implemented matrix-multiplication using OpenBLAS[3] function cblas_sgemm which also had a comparable performance with Intel MKL. These libraries highly optimizes the matrix-multiplication.

II. Performance Comparison

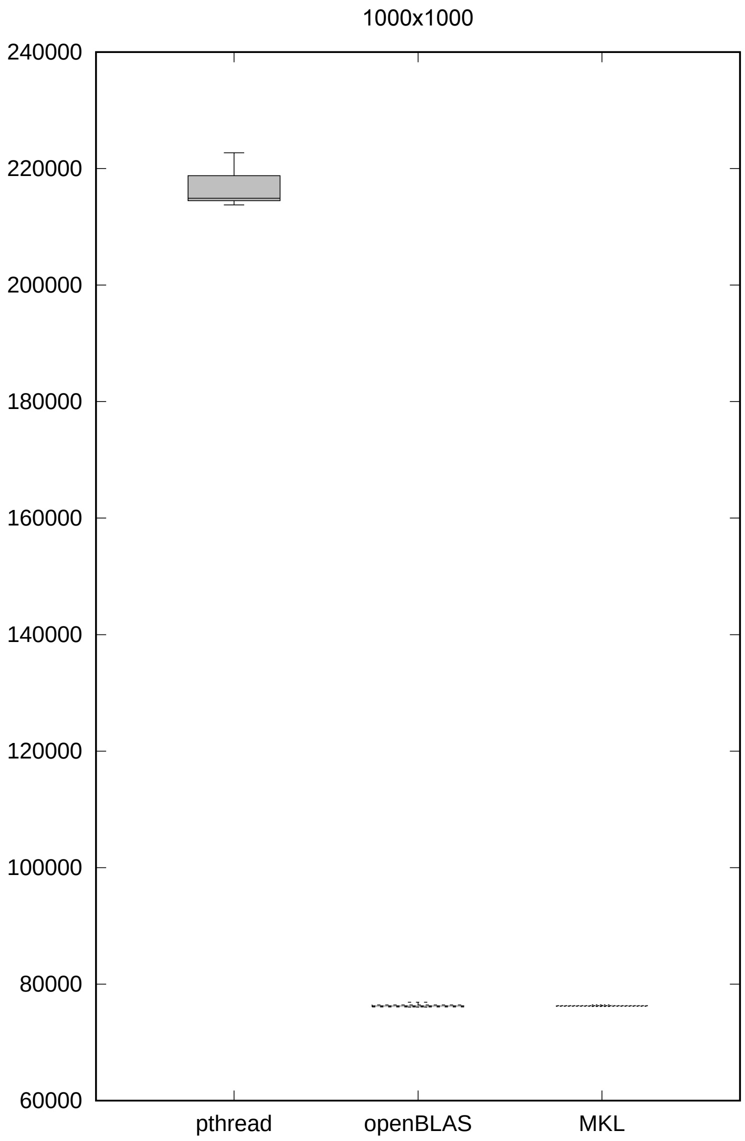
Mean of latencies over different matrix sizes

Size of Matrix	Open BLAS	MKL
500	2123	2155
750	4921	5001
900	6026	6578
1000	78078	79034

Standard Deviation of latencies over different matrix sizes

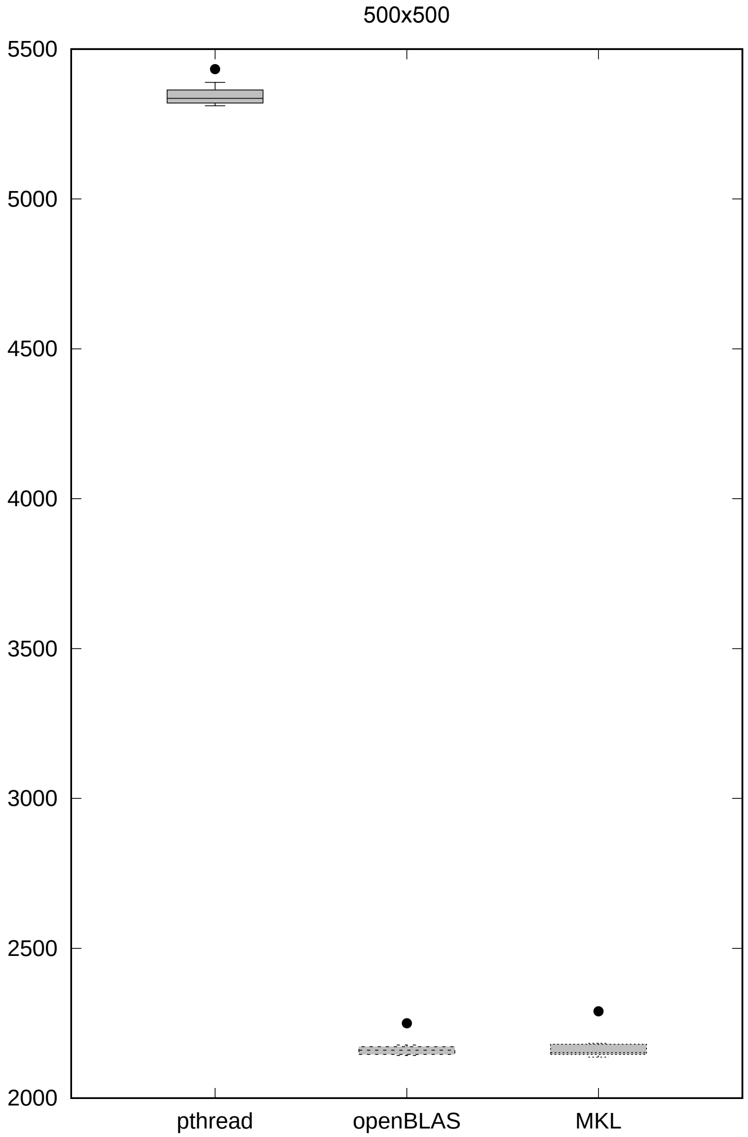
Size of Matrix	Open BLAS	MKL
500	11	27
750	134	53
900	19	51
1000	367	146

In the above tables with matrix size 1000 we used a kernel of 10 and a kernel size of 3 with others



III. Acceleration with pthreads

- In normal matrix we were carrying out single operation in one loop cycle.
- Using pthread we divided matrix into quarters and all 4 threads were computing the 4 parts in parallel, optimizing the performance compared to naive matrix-multiplication.
- Matrix-multiplication using pthread is nearly 3 times faster than normal matrix-multiplication.



References

[1] Intel, “Math Kernel Library,” `https://software.intel.com/en-us/mkl/choose-download/linux`, 2019.

[2] Intel, “Math Kernel Library,” `https://software.intel.com/en-us/articles/intel-mkl-link-line-advisor`, 2019.

[3] Xianyi, Z., “OpenBLAS,” `https://github.com/xianyi/OpenBLAS/wiki/User-Manual`, 2010.