

Performance Improvisation in Matrix Multiplication

Vijay, Vedant Bansal, Shivam
cs1170388@iitd.ac.in cs1170376@iitd.ac.in

17 February, 2019

1 Introduction

The Report consists details of different approaches(i.e libraries) used to improve the performance of LeNet structure. It helps to fasten the code and improve accuracy as well.

2 Linear Algebra Libraries

The main segment of LeNet algorithm which consumes time Convolution which solely depends on matrix multiplication. Direct Matrix multiplication by evaluation of each value proves out to be unacceptably slower for large number of values. Following are decribed the two improvement by usage of external libraries for C++ such as MKL and OpenBlas.

2.1 MKL

[1] Intel MKL greatly improves the performance of Basic Linear Algebra Subprograms (BLAS) and Linear Algebra Package (LAPACK) functions since it takes advantage of special features in the new generation of Intel processors that greatly speed up matrix operations. With Intel MKL, we don't need to modify our source code to take advantage of new features of Intel processors.

2.2 OpenBlas [2]

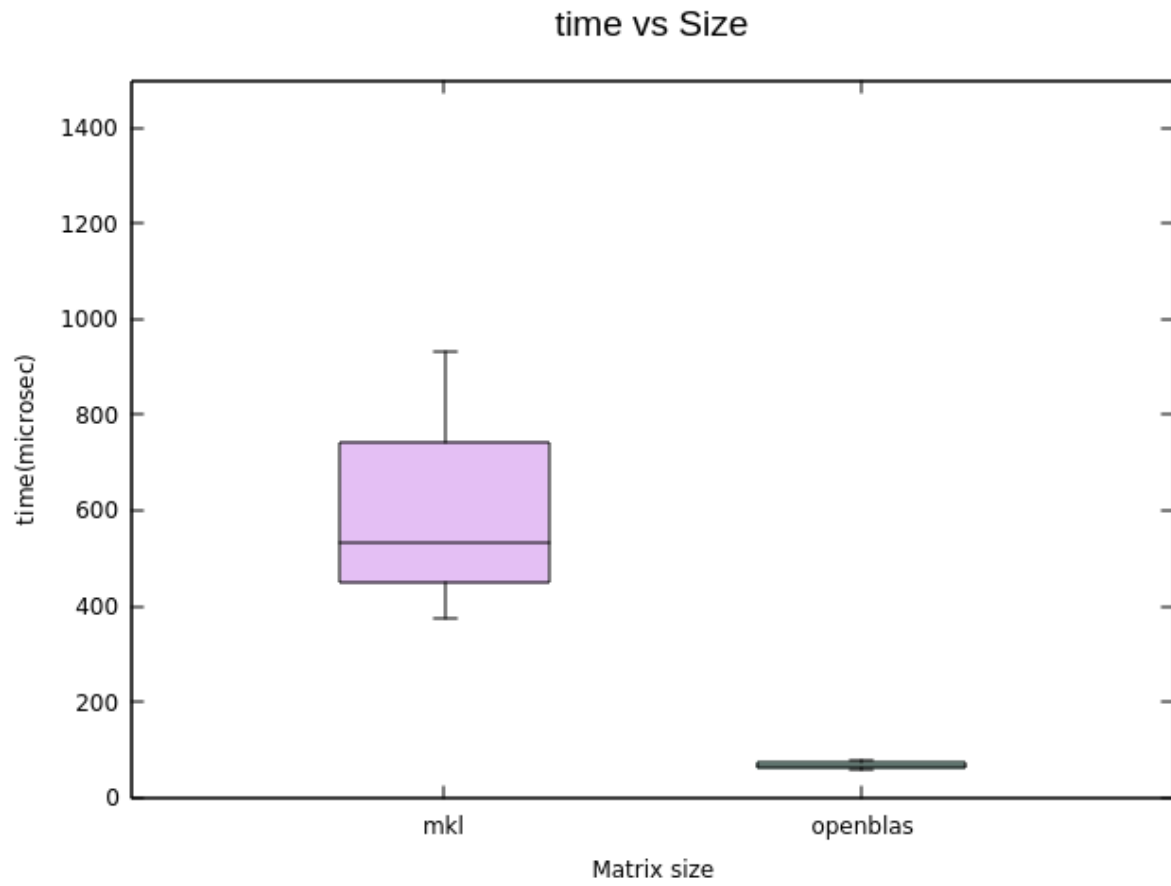
BLAS is divided into three levels:

- Level 1 defines a set of linear algebra functions that operate on vectors only. These functions benefit from vectorization.

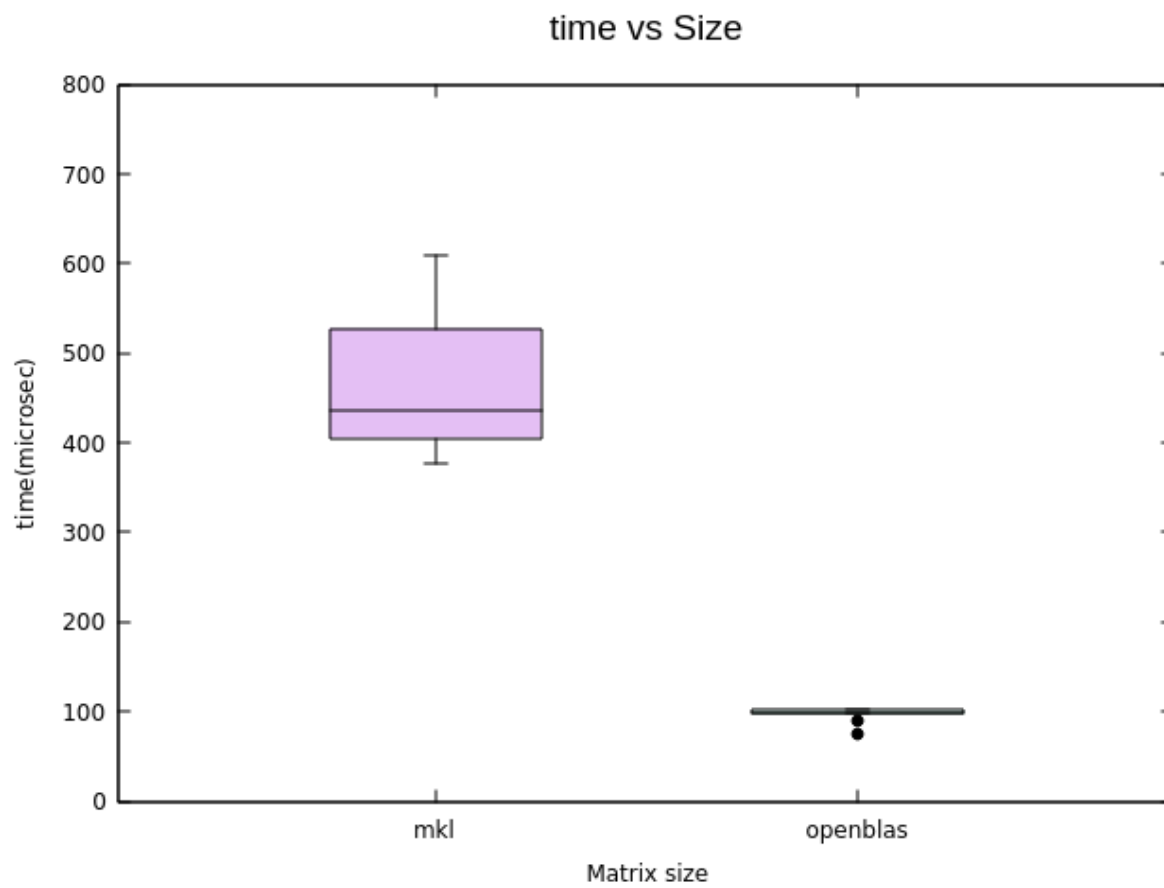
- Level 2 functions are matrix-vector operations, e.g. some matrix-vector product. These functions could be implemented in terms of Level1 functions. However, you can boost the performance of this functions if you can provide a dedicated implementation that makes use of some multiprocessor architecture with shared memory.
- Level 3 functions are operations like the matrix-matrix product. Again you could implement them in terms of Level2 functions. But Level3 functions perform $O(N^3)$ operations on $O(N^2)$ data. So if your platform has a cache hierarchy then you can boost performance if you provide a dedicated implementation that is cache optimized/cache friendly. The main boost of Level3 functions comes from cache optimization. This boost significantly exceeds the second boost from parallelism and other hardware optimizations.

3 Performance Comparision

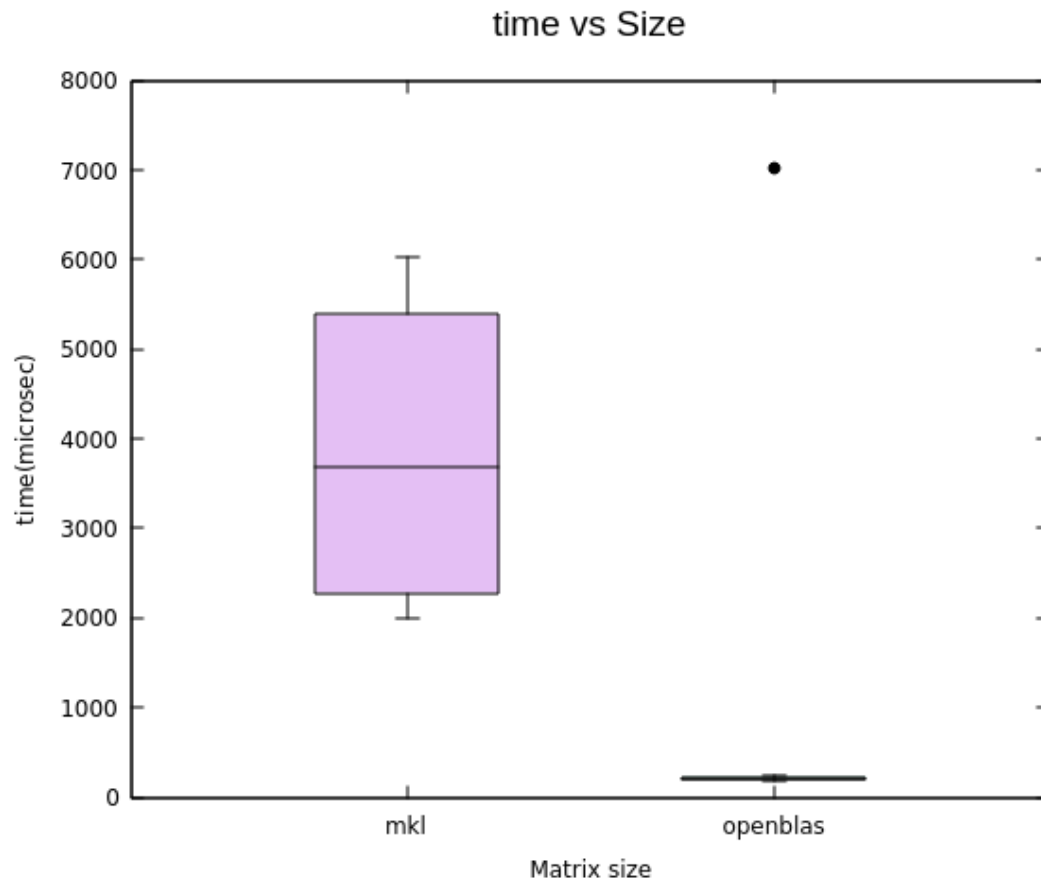
Following are three figures which depicts comparision between performance between OpenBlas and MKL library. It helps us understand the difference between two libraries by plotting time vs libarary curves for three sizes of Matrix mentioned below.



Size of matrix = 50



size of matrix = 75



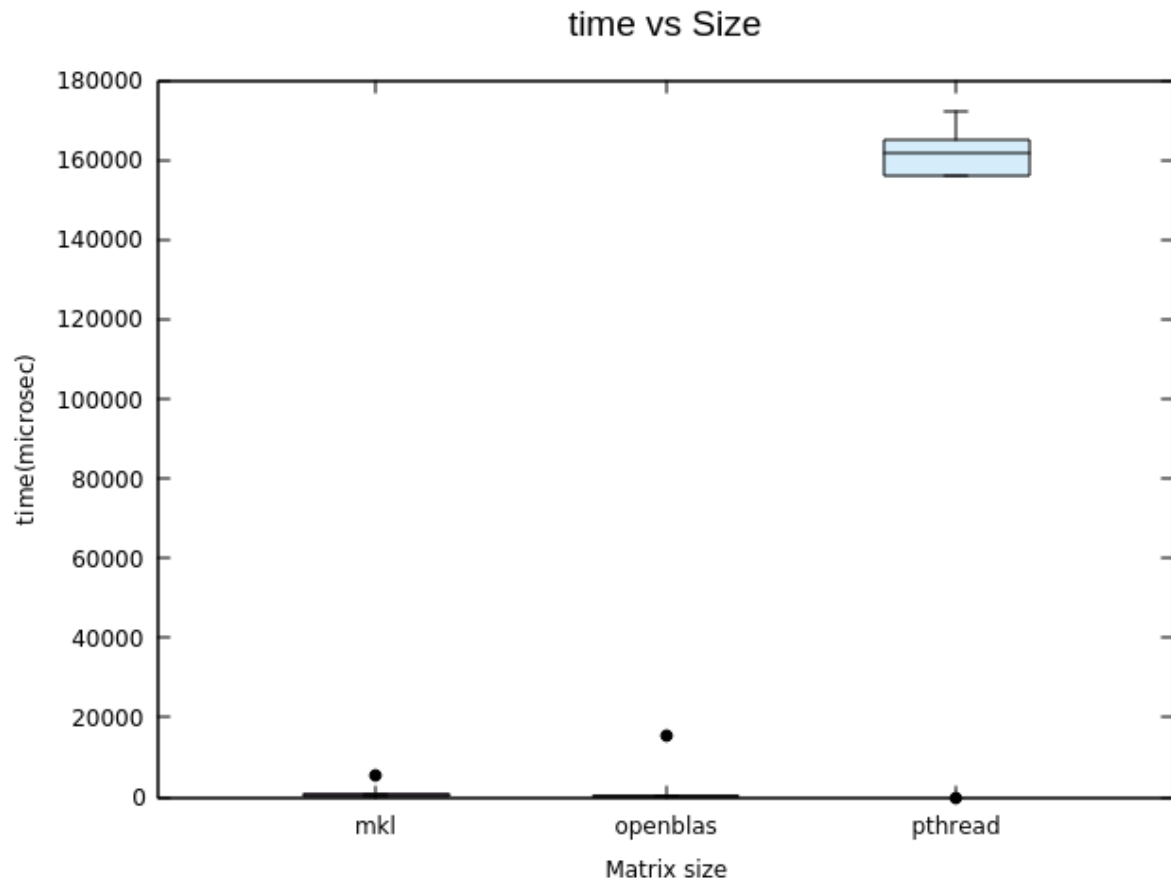
size of matrix =100

The overall comparison of MKL and OpenBlas is written in next section along with Pthreads.

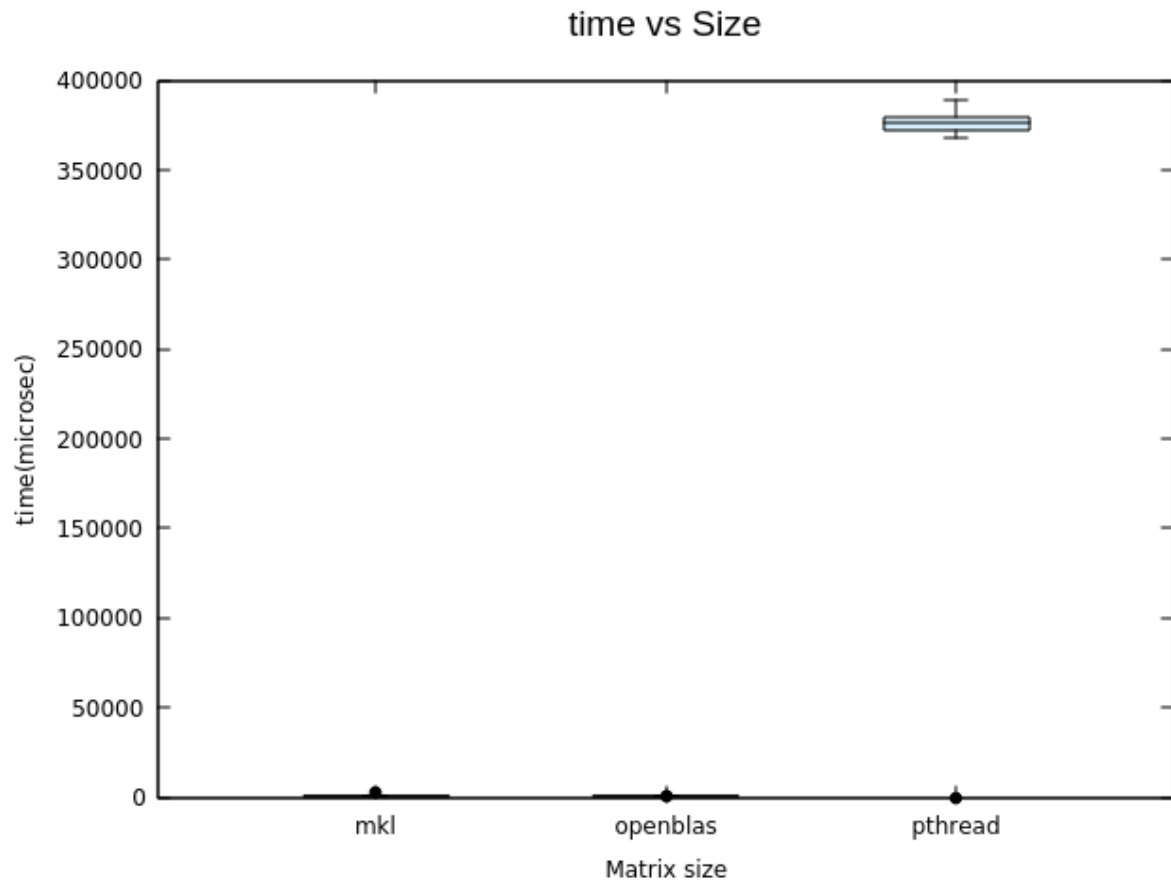
4 Acceleration with Pthreads

The work we did:

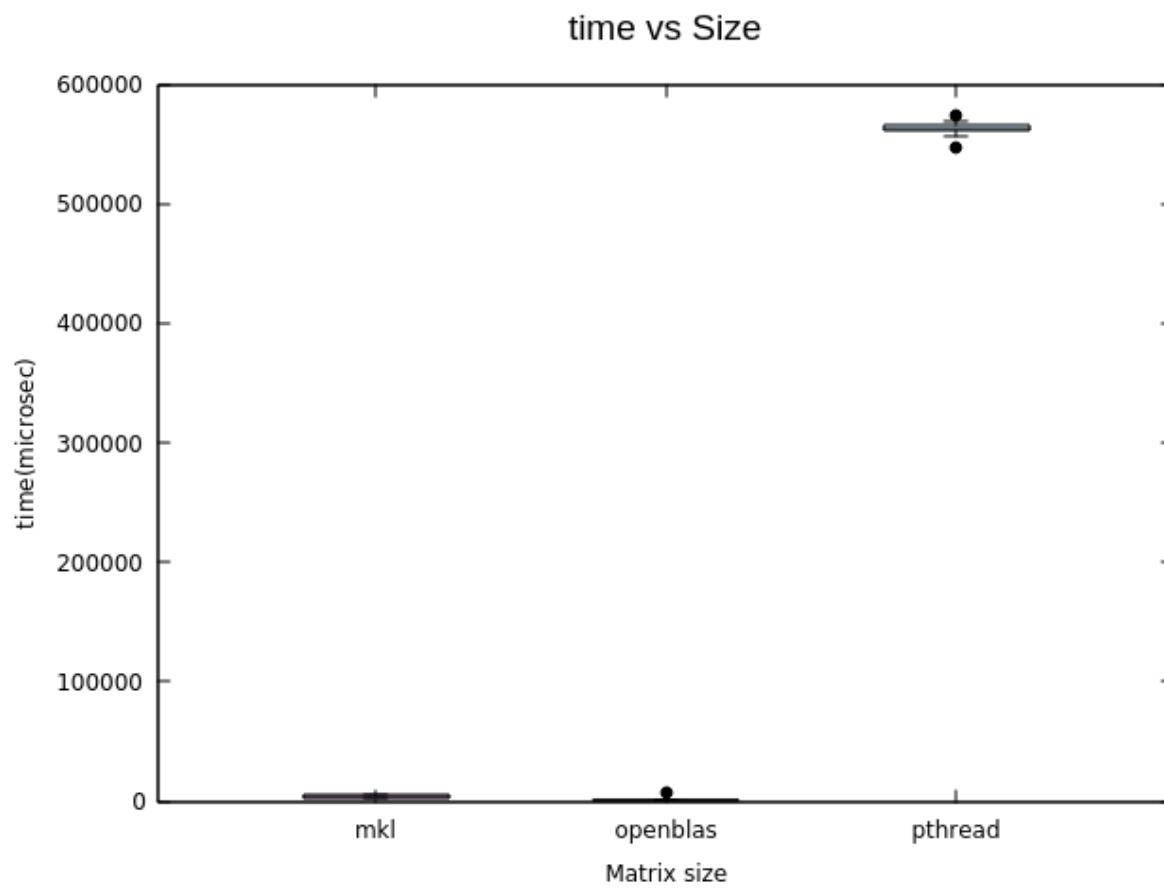
- We tried using 2 threads, 4 and 8 threads to increase the efficiency of the code.
- But increase in threads only lead to increase in run time.
- So we decided to go with MxN threads as we thought increase in threads must lead to decrease in runtime
- It turned out that it was not the case as creating new threads also takes some time and the advantage of creating new thread must be more than the time taken to create one in order to decrease the runtime.



Size of matrix = 50



size of matrix = 75



size of matrix =100

Comparision between MKL, OpenBlas and Pthreads:

Matrix Size	Kernel Size	Running Time- MKL		Running Time- OpenBlas		Running Time- Pthreads	
		x	d	x	d	x	d
50x50	3x3	1066.3	1470.6	1621.7	4666.7	161993.3	4855.2
75x75	3x3	656.5	606	99.2	3.48	377160.6	5564.1
100x100	3x3	3964	1520.1	888.5	2045.8	563411	7112.5

x: Mean Time(in microsecond)

d: Standard Deviation(in microsecond)

5 Conclusion

Time taken by MKL is less than OpenBlas is less than pthread. Pthread takes more time than others as we have used MxN threads and creating new thread is time consuming. MKL takes less time as we ran the code on an intel machine and MKL is optimized for intel processors whereas openblas is indifferent for the type of processors.

References

- [1] Khang T (Intel) Nguyen. Improving performance of math functions with intel® math kernel library.
["https://software.intel.com/en-us/articles/improving-performance-of-math-functions-with-intel-math-kernel-libr"](https://software.intel.com/en-us/articles/improving-performance-of-math-functions-with-intel-math-kernel-libr)
 Date Accessed: 18 February 2019.
- [2] StackOverflow DeusAduro. Blas performance.
["https://stackoverflow.com/questions/1303182/how-does-blas-get-such-extreme-performance"](https://stackoverflow.com/questions/1303182/how-does-blas-get-such-extreme-performance).
 Date Accessed: 18 February 2019.