BOOST MATRIX MULTIPLICATION

Shivam : 2017CS10377
$e$mail: cs1170377@iitd.ac.in
Meenal : 2017CS10351
$e$mail: cs1170351@iitd.ac.in

February 17, 2019

1 Linear Algebra Libraries

There exists various Linear Algebra Libraries which can be used to enhance the speed of computation. In our subtask, we have used following two libraries to accelerate matrix-matrix multiplication.

1.1 MKL

Intel MKL performs fast matrix-matrix multiplication.It provides a compiler flag to guarantee that the fastest code path is used at runtime.
In our code we have included the library "mkl.h" and the function cblas sgemm for performing multiplication through mkl.
We have used it from [1].

1.2 OpenbLAS

OpenBLAS is an optimized BLAS library. OpenBLAS adds optimized implementations of linear algebra kernels for several processor architectures. Its performance is comparable(slightly slow) to Intel MKL.
In our code we have included the library "cblas.h" and the function cblas sgemm for performing multiplication through openBLAS.
We have used it from [2].

2 Performance Comparison

Matrix Size	MKL (Average)	MKL (Std. Deviation)	OpenBLAS (Average)	OpenBLAS (Std. Deviation)
60	10.2	12.76	19.8	10.12
100	22.0	11.51	28.6	12.42
140	44.6	23.79	52.0	5.74
180	59.8	18.32	62.2	17.15

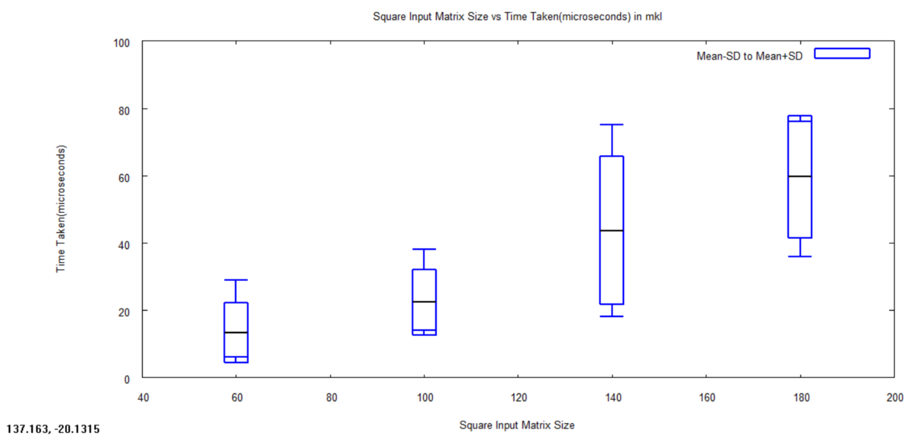

Figure 1: Plot for mkl

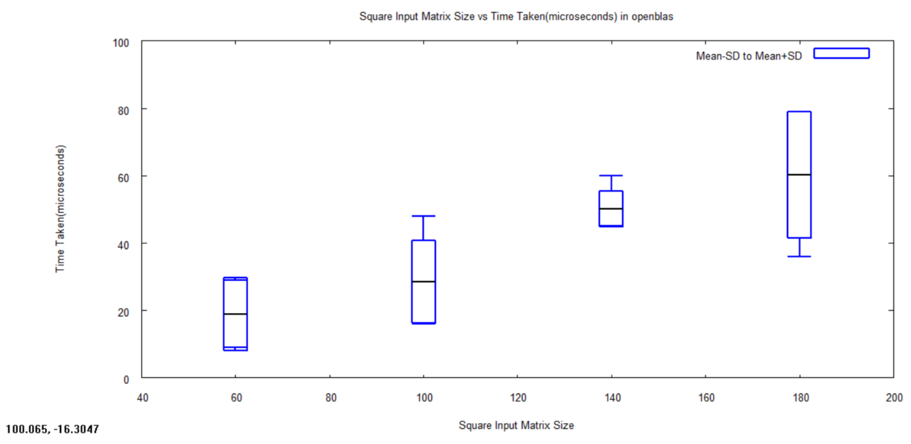

Figure 2: Plot for openblas

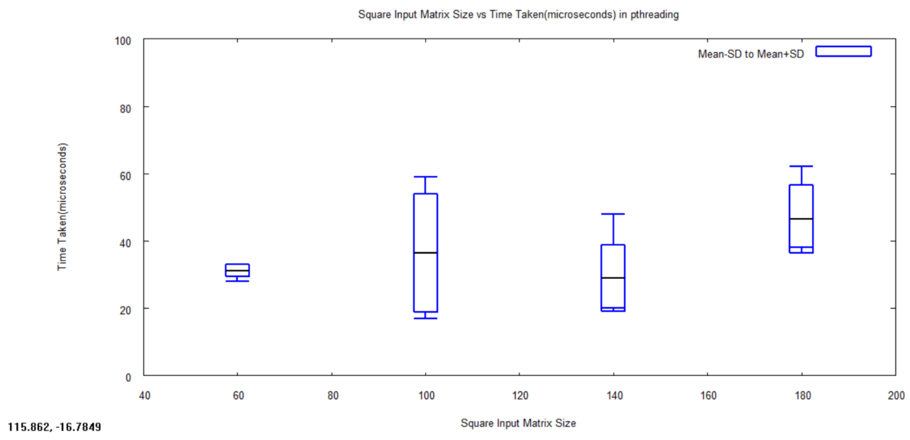

Figure 3: Plot for Pthread

3 Acceleration With Pthreads

- POSIX thread (pthread) allows use of concurrent or parallel programming to get computed results with faster speeds.
- We implemented matrix multiplication used in convolution using pthreads. We worked with 4 threads,on all matrix sizes mentioned in the table below.
- In our implementation different threads compute different parts of the resultant matrix and then are joined to get the final output.
- Sometimes we were facing the issue that our program exited without waiting for the last thread to complete, and the other times the program exited with complete and correct output. We used join() function to join pthreads but could not figure out what is wrong.
- Pthread took a bit longer on smaller matrices accounting to the time taken in creating and destroying multiple threads.

Matrix Size	Original (Average)	Original (Std. Deviation)	Pthread (Average)	Pthread (Std. Deviation)
60	18.4	12.54	31.2	1.96
100	40.4	11.52	38.2	17.91
140	32.4	15.48	28.6	11.36
180	48.4	16.71	46.8	10.12

References

[1] Intel. Developer zone: Math kernel library. `"https://software.intel.com/en-us/mkl"`. Date Accessed: 21 January 2019.

[2] Zhang Xianyi. xianyi: github. `"https://github.com/xianyi/OpenBLAS"`. Date Accessed: 23 January 2019.