SiWiR1 Assignment
Optimization of Matrix Multiplication

Group 10
Members :- Raju Ram, Sagar Dolas, Seyedehelmira Birang Oskouei

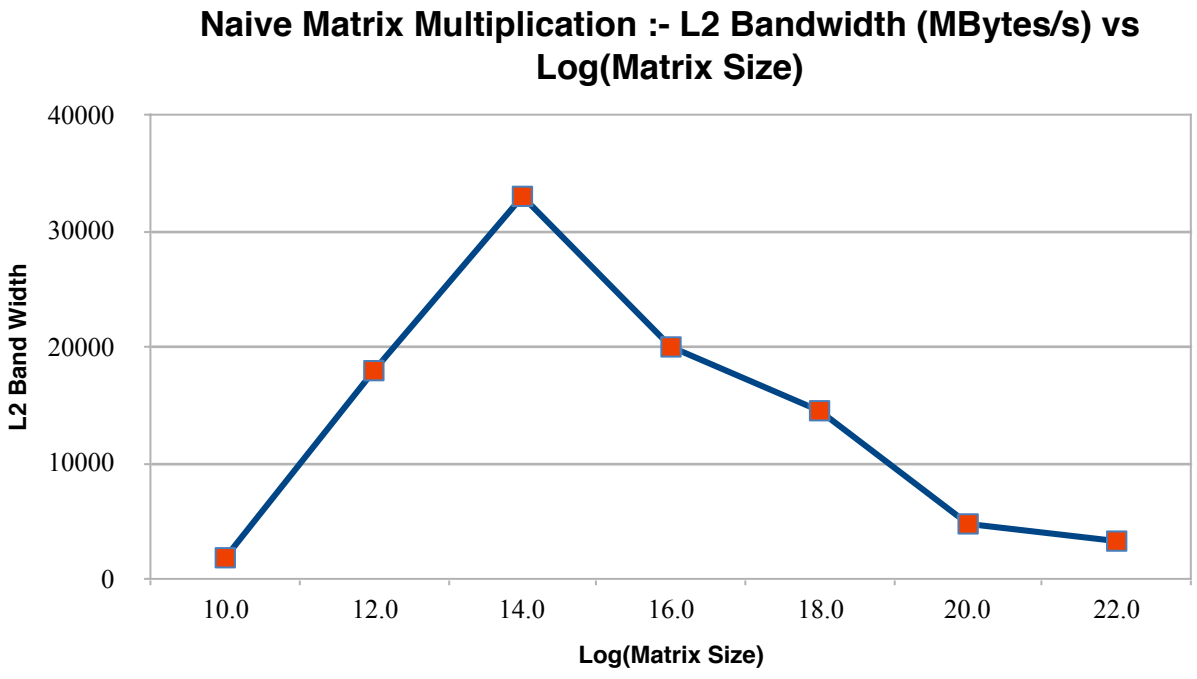
We have used the naive matrix multiplication to perform C = A*B . Matrix A and B are stored in one dimensional array and the entries are read row-wise. We have optimized the matrix multiplication with Blocking and with Copy Optimization Technique.

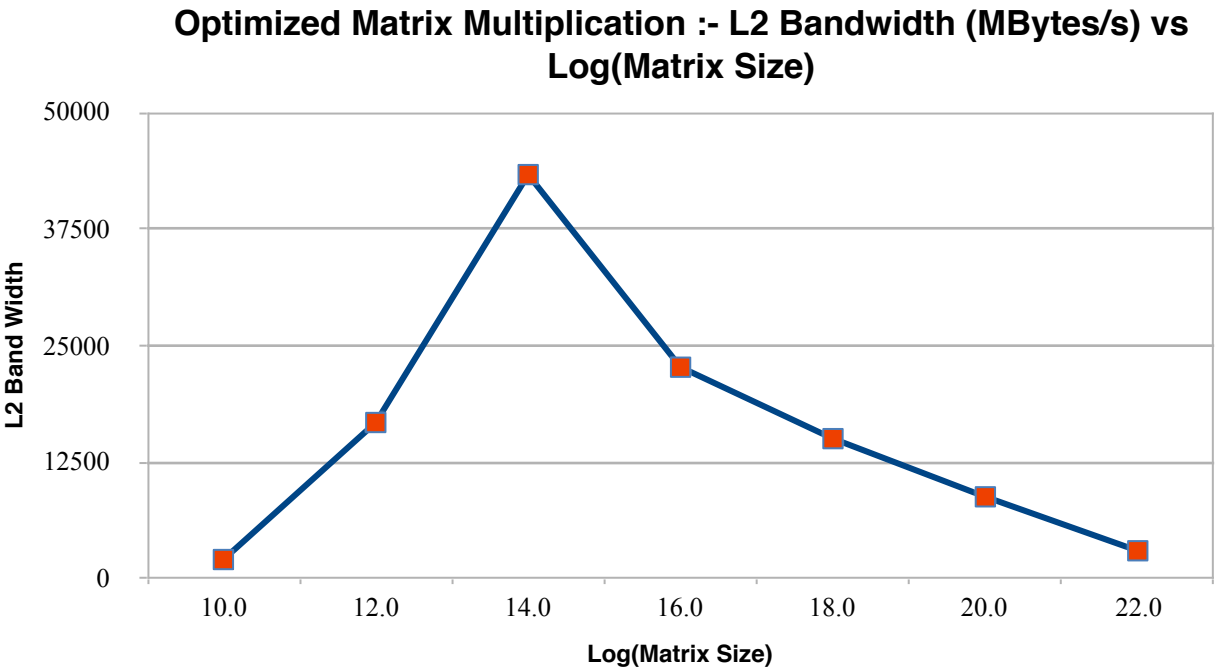
Blocking Technique :- In this technique the matrix is divided into various blocks and matrix multiplication is performed block by block, since at the time of computation only block elements are there in the Cache, it minimises Cache misses.

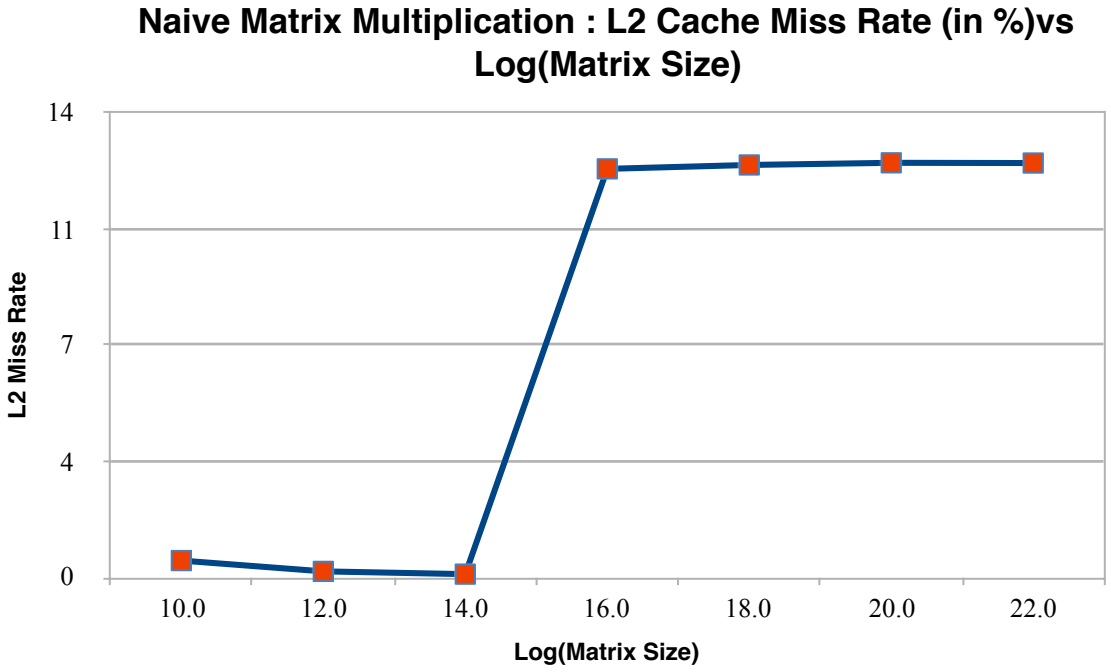
Copy Optimization:- When we multiply A with B, The entries of A are read row wise and entries of B are read column wise. Since A and B are stored in row major format, we can access elements of matrix A without memory jump but reading of elements of B causes unnecessary memory jump and which makes the program inefficient so we have saved B in column major format so that it does not cause unnecessary memory jump.

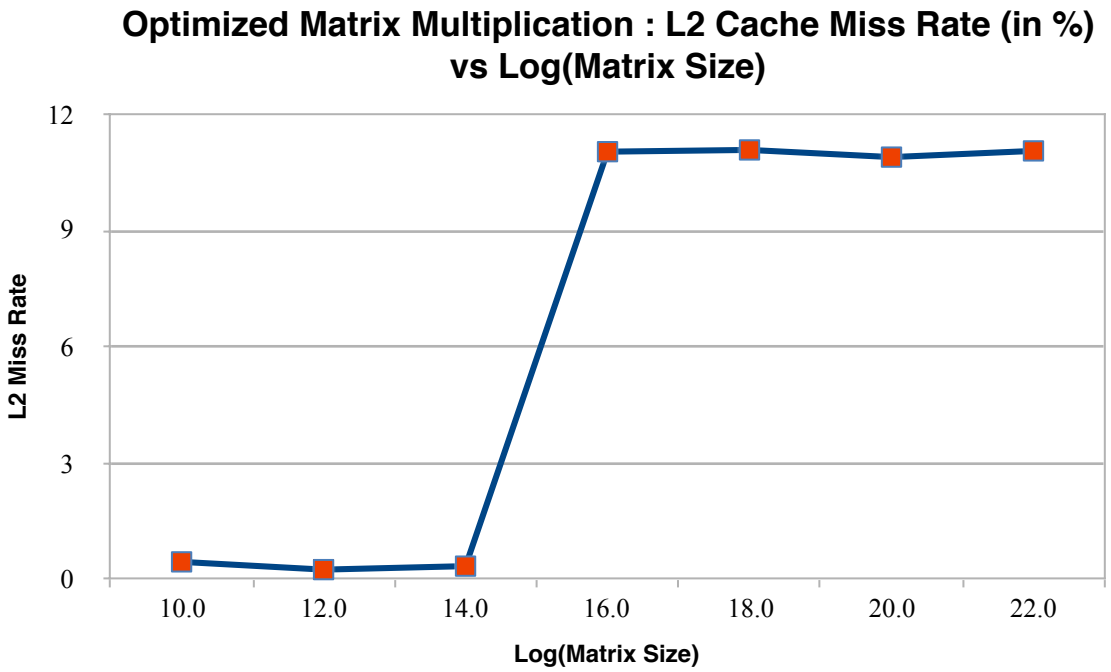
We have also analysed the run time, L2 bandwidth, L2 Cache miss rate and double precision FLOPS for native matrix multiplication and optimised matrix multiplication.

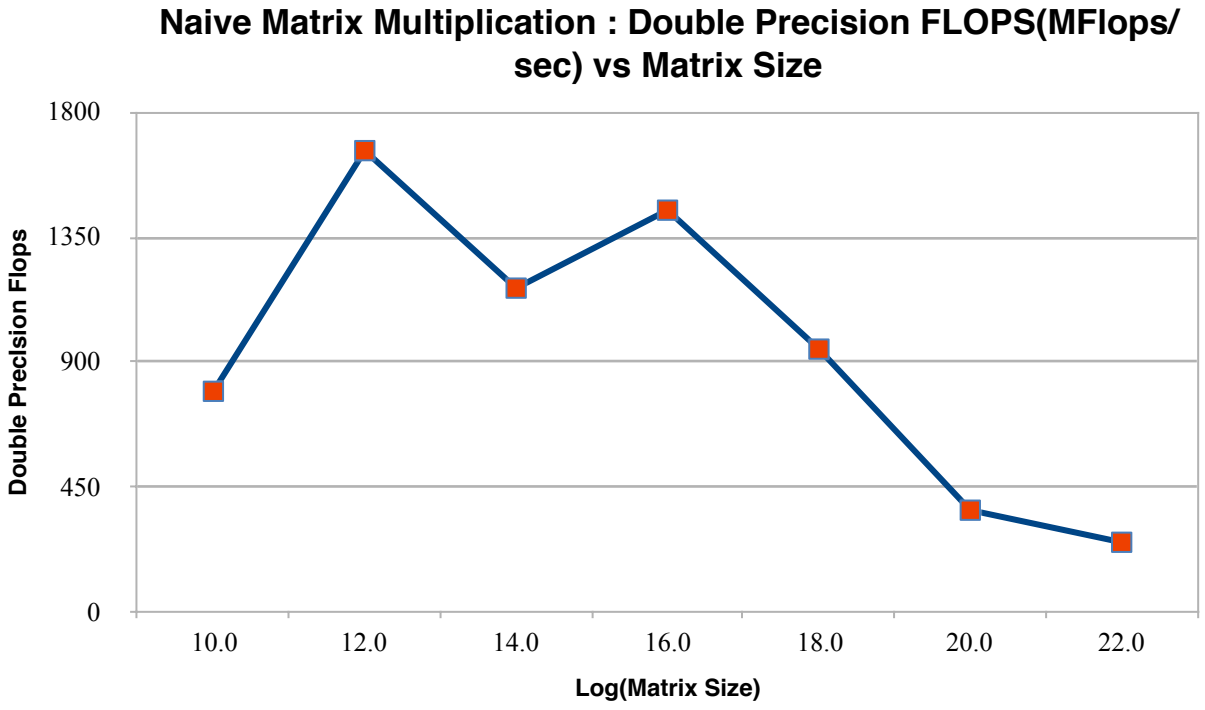
We have plotted above parameters for various matrixes; abscissa is the log of matrix.
Ex :- for 32*32 matrix size of matrix is 1024 and we take log(1024) = 10.

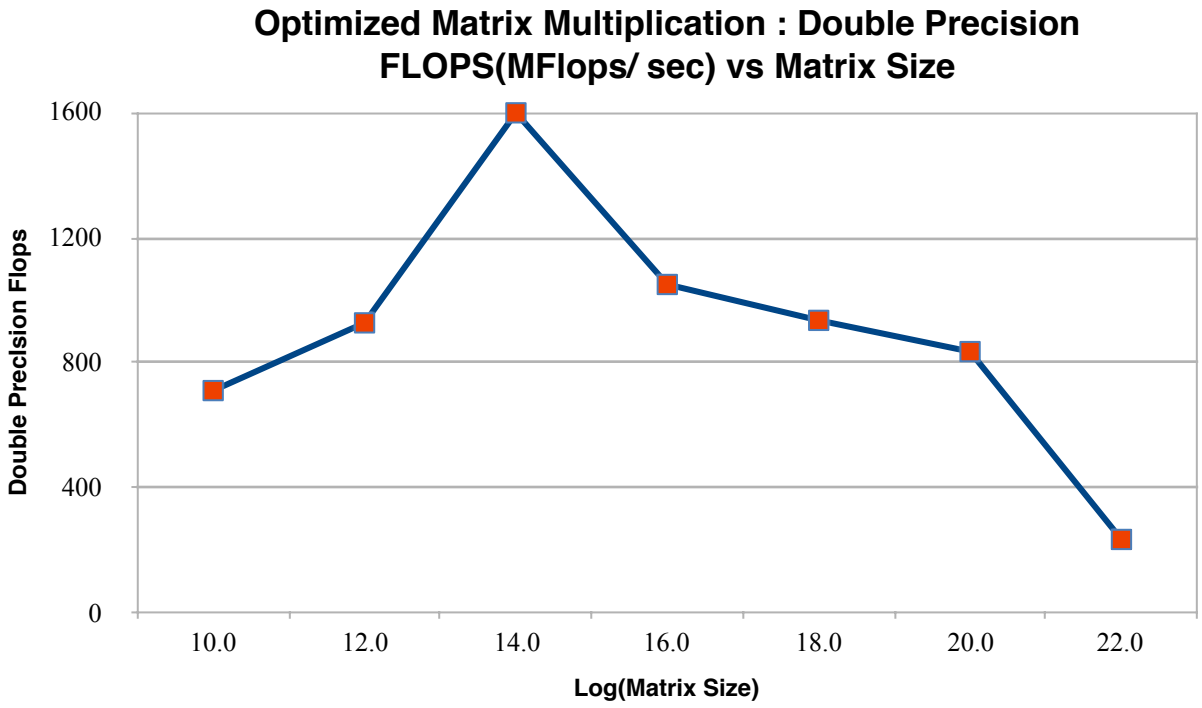


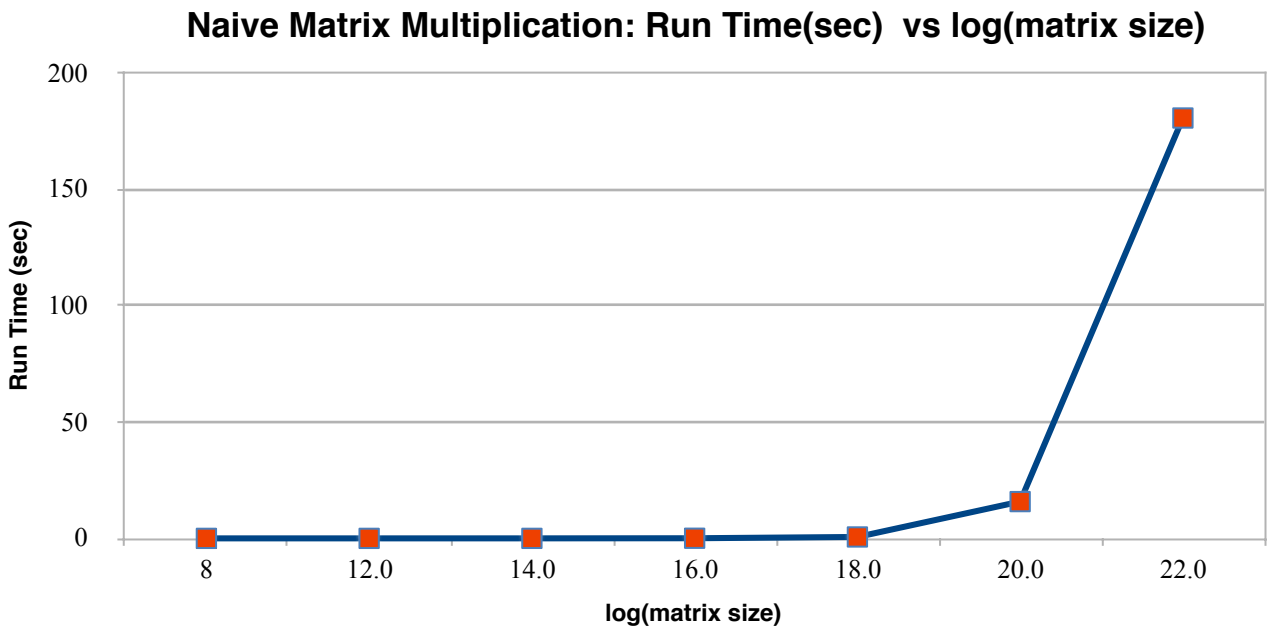

Optimized Matrix Multiplication :- L2 Bandwidth (MBytes/s) vs Log(Matrix Size)
L2 Band Width
50000
37500
25000
12500
0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
Log(Matrix Size)

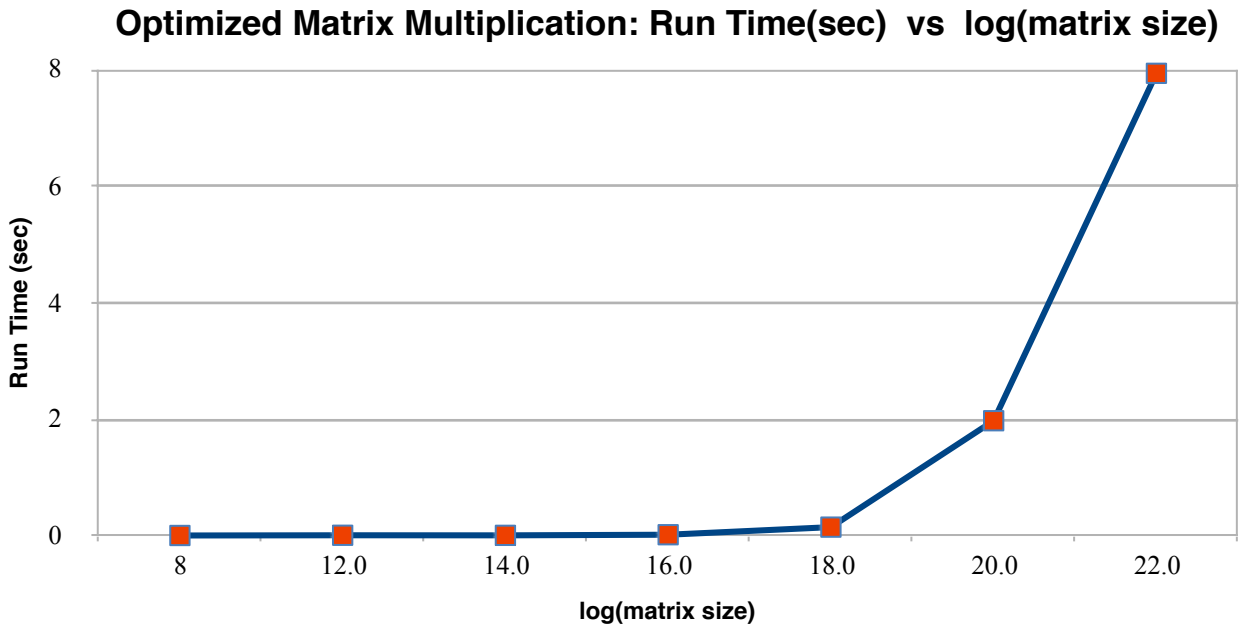

Naive Matrix Multiplication : L2 Cache Miss Rate (in %)vs Log(Matrix Size)
L2 Miss Rate
14
11
7
4
0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
Log(Matrix Size)


Optimized Matrix Multiplication : L2 Cache Miss Rate (in %) vs Log(Matrix Size)
L2 Miss Rate
12
9
6
3
0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
Log(Matrix Size)


Naive Matrix Multiplication : Double Precision FLOPS(MFlops/ sec) vs Matrix Size
Double Precision Flops
1800
1350
900
450
0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
Log(Matrix Size)


Optimized Matrix Multiplication : Double Precision
FLOPS(MFlops/ sec) vs Matrix Size
Double Preclsion Flops
1600
1200
800
400
0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
Log(Matrix Size)


Naive Matrix Multiplication: Run Time(sec) vs log(matrix size)
Run Time (sec)
200
150
100
50
0
8
12.0
14.0
16.0
18.0
20.0
22.0
log(matrix size)



From the above graph we observe increment in the performance for optimisation technique since time taken by naive matrix multiplication was about 180 seconds whereas the time taken by efficient implementation is about 8 seconds for 1024*1024 matrix multiplication.