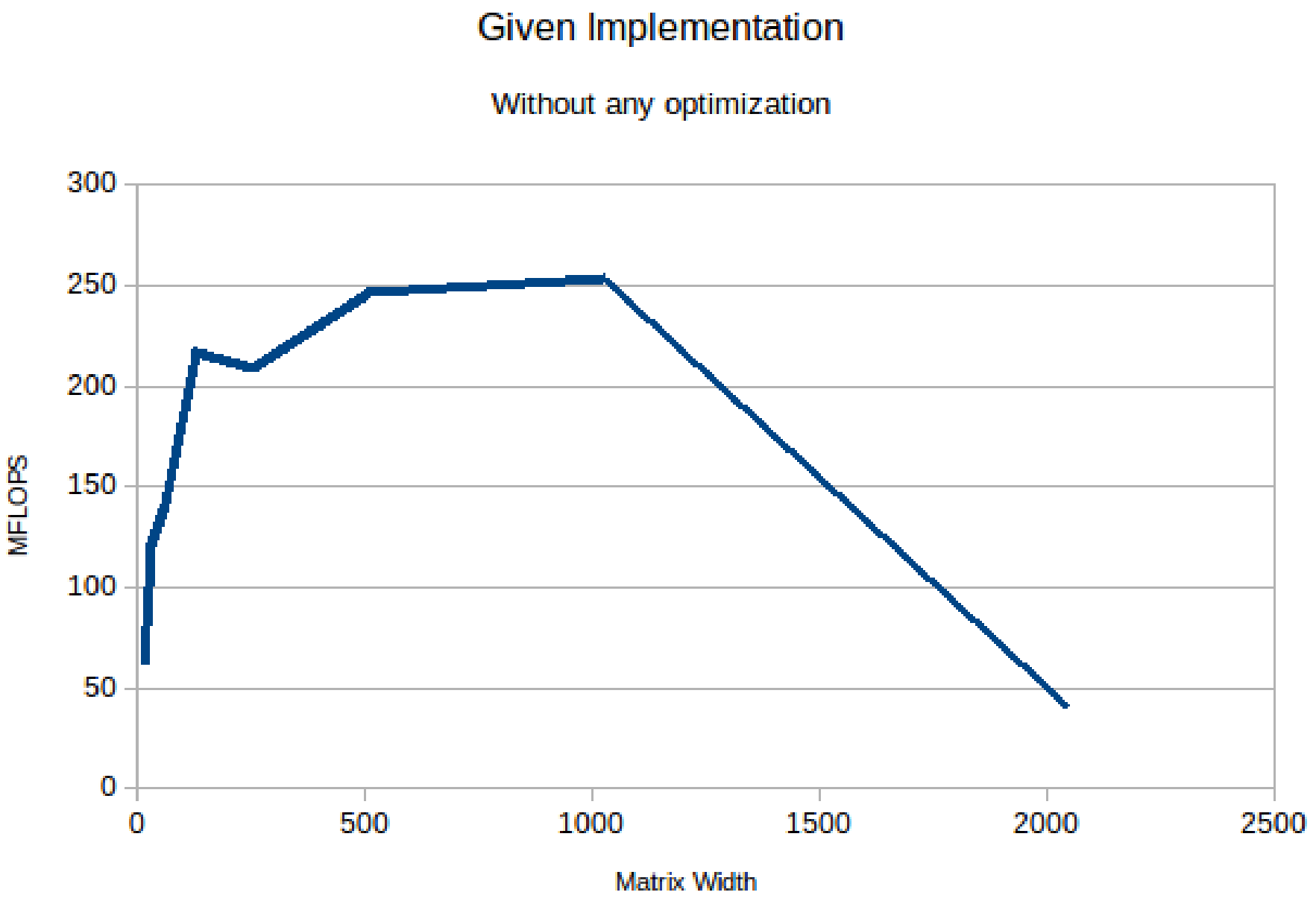
Matrix Multiplication Observations

Gaurav Kukreja
Evangelos Drossos

Given Implementation



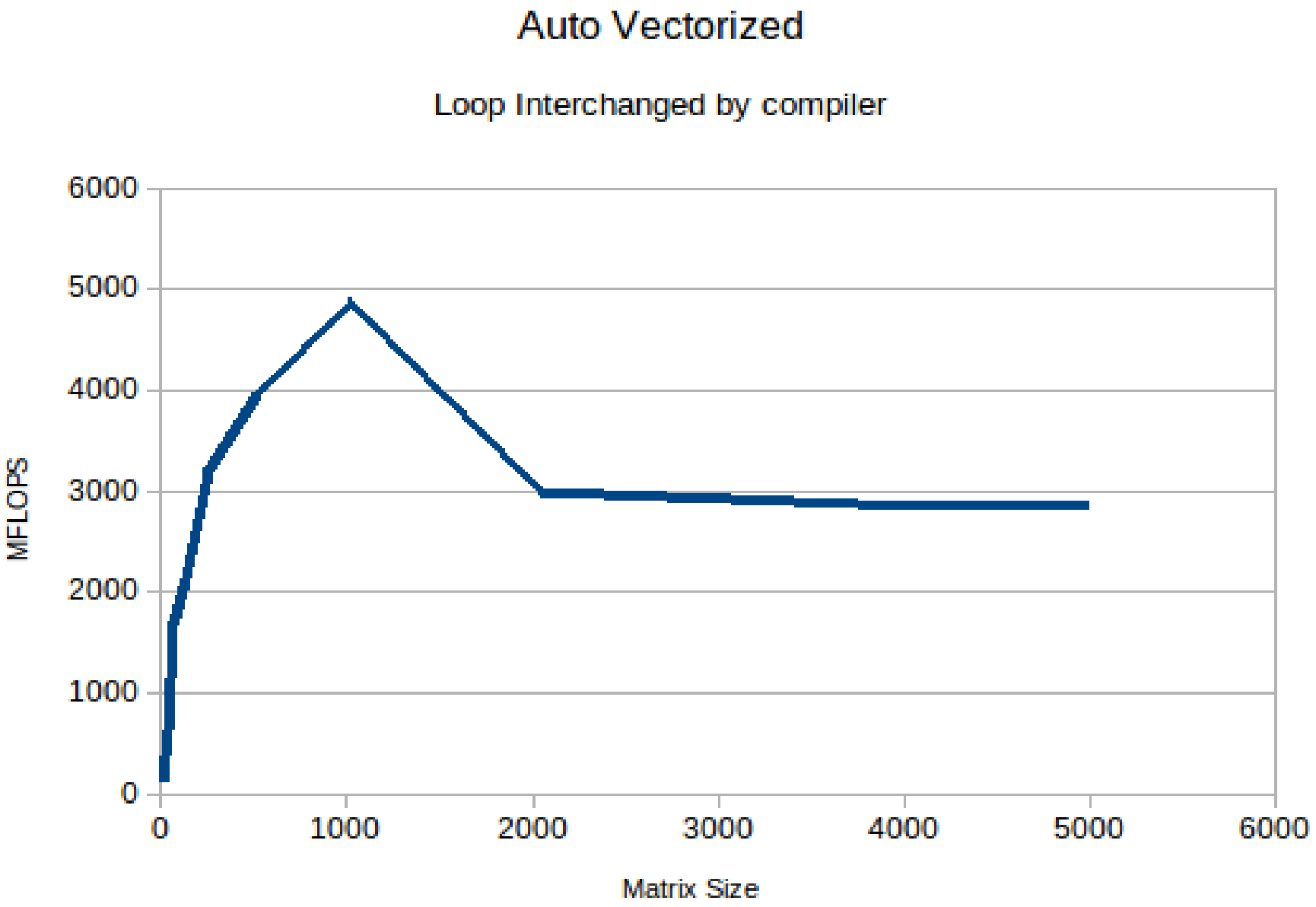
Given Implementation

- L1 Cache 32 KB
- Can store two matrix of 63*63
- Steep rise of performance, until matrix size 63. Matrix fits in L1 Cache

Given Implementation

- L3 Cache 20 MB
- Can store two matrix of 1140*1140
- Decent performance until matrix size 1140, followed by sudden dip.
- The dip is caused due to L3 Misses.

Autovectorized (O3)



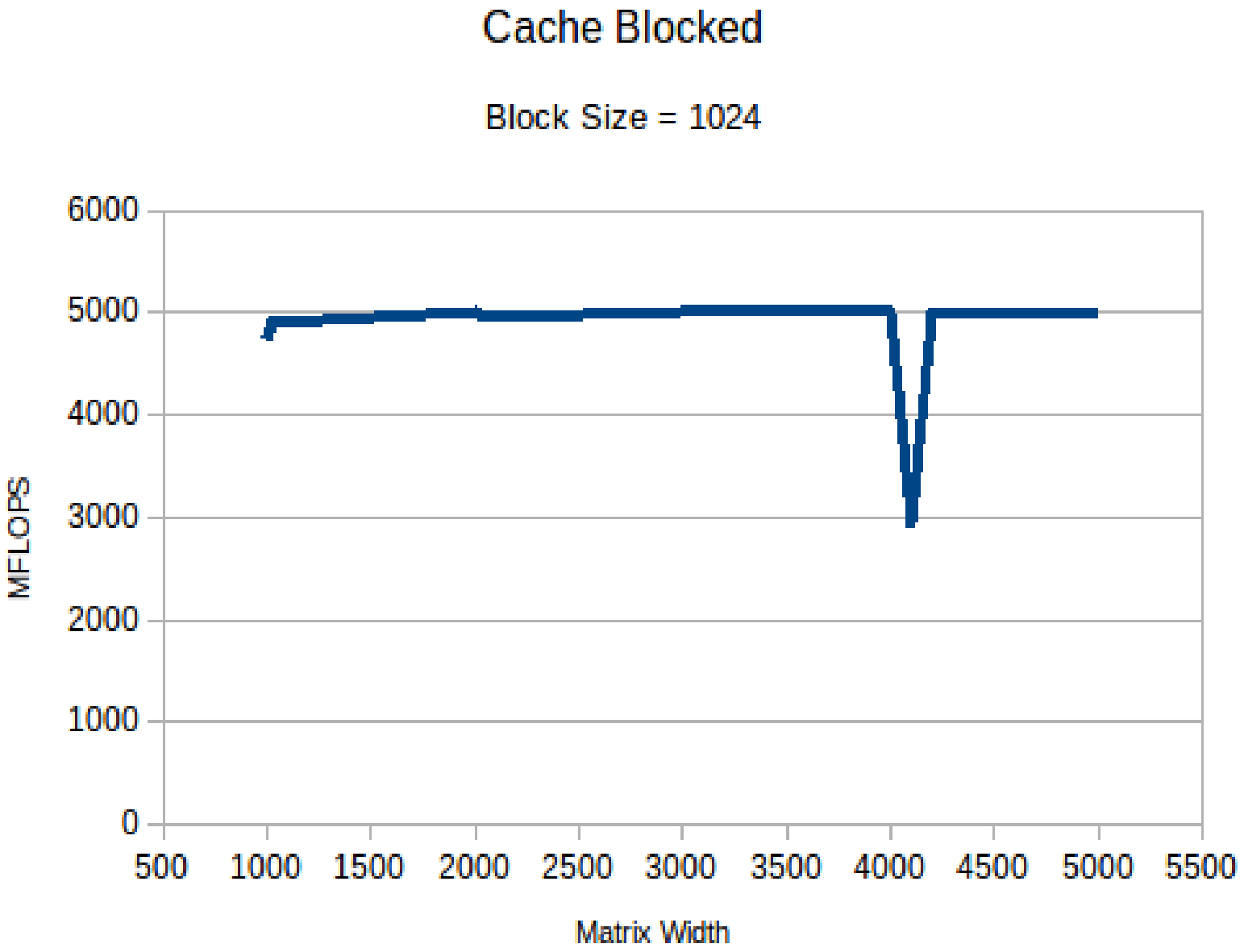
Autovectorized

- L1 Cache 32 KB
- With the loop interchanging, the L1 is better optimized, even for a matrix of size 128.
- Steep Improvement is noticed.
- Note, that MFLOPS is much much higher than naïve implementation, owing to vectorization and other optimizations by compiler.

Autovectorized

- L3 Cache 20 MB
- Again, we can see a performance dip beyond matrix size 1024.
- This is because of L3 misses.

Cache Blocked (block_size = 1024)



Cache Blocked (block_size = 1024)

- Block size has been chosen as 1024. Implementation is optimized for L3 cache access.
- Performance same at 1024, compared to Autovectorized.
- Performance remains consistent beyond matrix with 1024.

Cache Blocked (block_size = 1024)

- Sudden dip at 4096, could be due to cache line collisions. ???