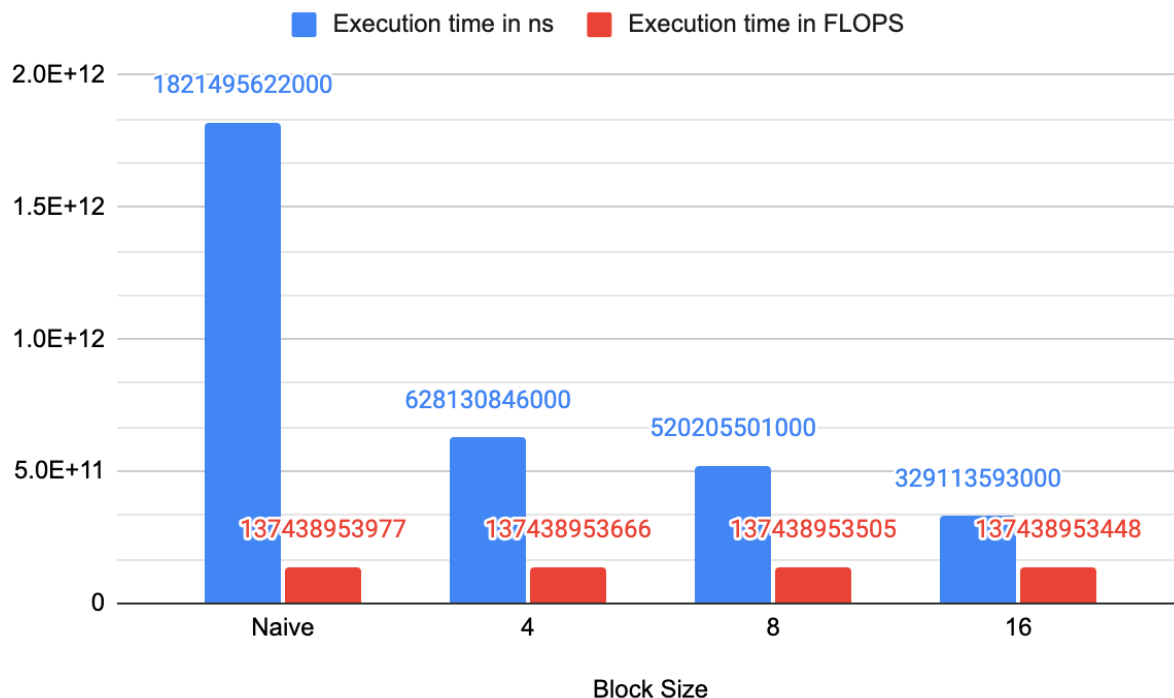


Problem 1: Matrix Multiplication

The following is the results of running a naive matrix multiplication and block matrix multiplication with block sizes of 4, 8, and 16 of two 4096*4096 sized matrices.

The figures below (graph and table) depict the corresponding execution times in ns and number of FLOPS for each block size. As evident from the graph, it can be seen that as we utilize block matrix multiplication and increasing the size of the blocks (b), the execution has decreased linearly. On the other hand, the execution time in FLOPS remained almost constant amongst the four blocks.

Block size	Execution time in ns	Execution time in FLOPS
Naive	1821495622000	137438953977
4	628130846000	137438953666
8	520205501000	137438953505
16	329113593000	137438953448

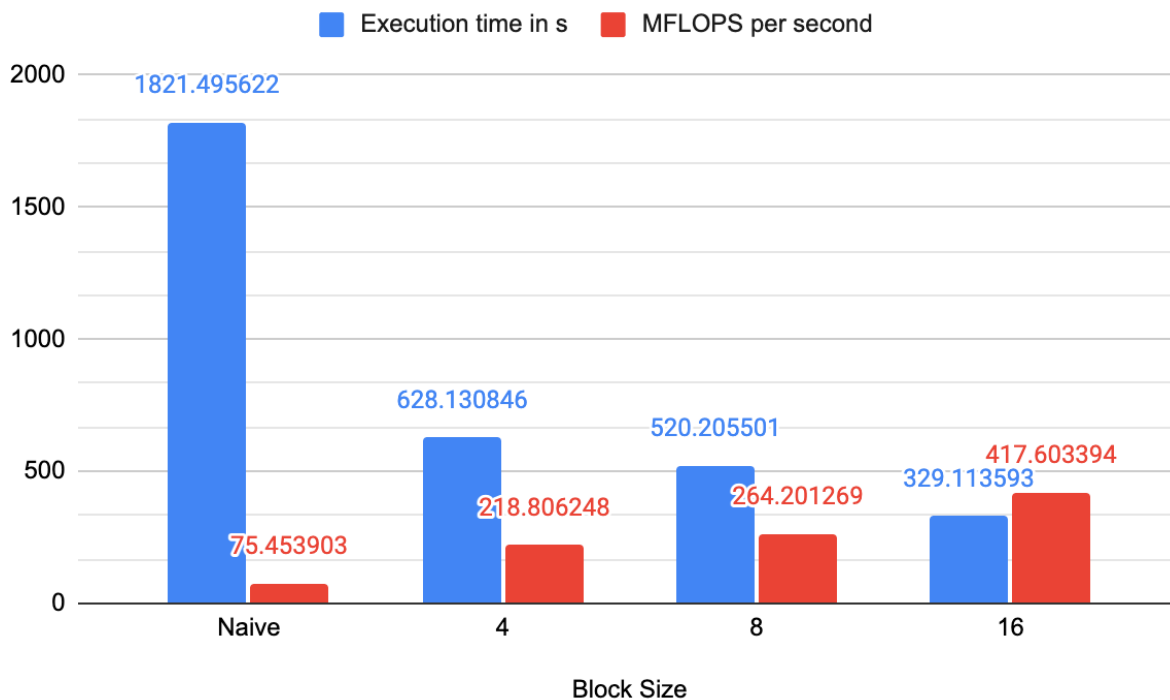


The figures below (graph and table) depict the corresponding execution times in s and the number of MFLOPS per second for each block size. As consistent with the graph above, the

Tingqi (Ting) Wang

execution time linearly decreases as we increase the block size. In contrast, it can be seen that the MFLOPs per second is increasing linearly.

Block size	Execution time in s	MFLOPS per second
Naive	1821.495622	75.453903
4	628.130846	218.806248
8	520.205501	264.201269
16	329.113593	417.603394



Terminal logs:

```
tingqi wang@tts-macbook-pro EE_451_S_2023_PHW_1 % ./p1b 4
Number of FLOPs = 0, Execution time = 628.130846 sec,
218.806248 MFLOPs per sec
C[100][100]=879616000.000000
tingqi wang@tts-macbook-pro EE_451_S_2023_PHW_1 % ./p1b 8
Number of FLOPs = 0, Execution time = 520.205501 sec,
264.201269 MFLOPs per sec
C[100][100]=879616000.000000
tingqi wang@tts-macbook-pro EE_451_S_2023_PHW_1 % ./p1b 16
Number of FLOPs = 0, Execution time = 329.113593 sec,
417.603394 MFLOPs per sec
```

Tingqi (Ting) Wang

```
C[100][100]=879616000.000000
```

```
tingqi wang@tts-macbook-pro EE_451_S_2023_PHW_1 % ./p1a
```

```
Number of FLOPs = 0, Execution time = 1821.495622 sec,
```

```
75.453903 MFLOPs per sec
```

```
C[100][100]=879616000.000000
```

Only 1 run for each of the four block sizes were run as each run took a substantial amount of time and there was limited time at hand.

My test platform was a Macbook Pro (macOS Ventura 13.5.1 (22G90)) with the following configurations:



Problem 2: K-Means Algorithm

The following is the result of running a simplified K-means algorithm on a raw image input and outputting a raw image output after 30 iterations (instead of standard procedure to achieve convergence).

The runtime of the program was 0.294012 sec (excluding read and write time).

Tingqi (Ting) Wang

The image below on the left is the raw image input and the image below on the right is the raw image output (both of the images were generated using imageJ)



Terminal Logs:

tingqi wang@tts-macbook-pro EE_451_S_2023_PHW_1 % ./p2

Time taken to run: 0.294012 sec