

Homework 1

Name: Anand Ravi

System Properties:

Processor : Intel Core i3-6100U

Number of Cores: 4

Operating System: Ubuntu 16.04

Problem 1a: Naive Multiplication

Number of Runs : 5

Average execution time = $(1408 + 1351 + 1372 + 1508 + 1397) / 5 = 1407 \text{ sec} = 1.41\text{e}+12 \text{ ns}$

Average performance = $(97.6 + 101 + 100 + 91 + 98) / 5 = 97.52 \text{ MFLOPS/sec}$

Problem 1b: Block Matrix Multiplication

Case 1 : Block Size = 4

Number of Runs = 2

Average execution time = $(575 + 560) / 2 = 567.5 \text{ Seconds}$

Average performance = $(245 + 273) / 2 = 259 \text{ MFLOPS/sec}$

Case 2 : Block Size = 8

Number of Runs = 2

Average execution time = $(502 + 504) / 2 = 503 \text{ Seconds}$

Average performance = $(273 + 272) / 2 = 273.5 \text{ MFLOPS/sec}$

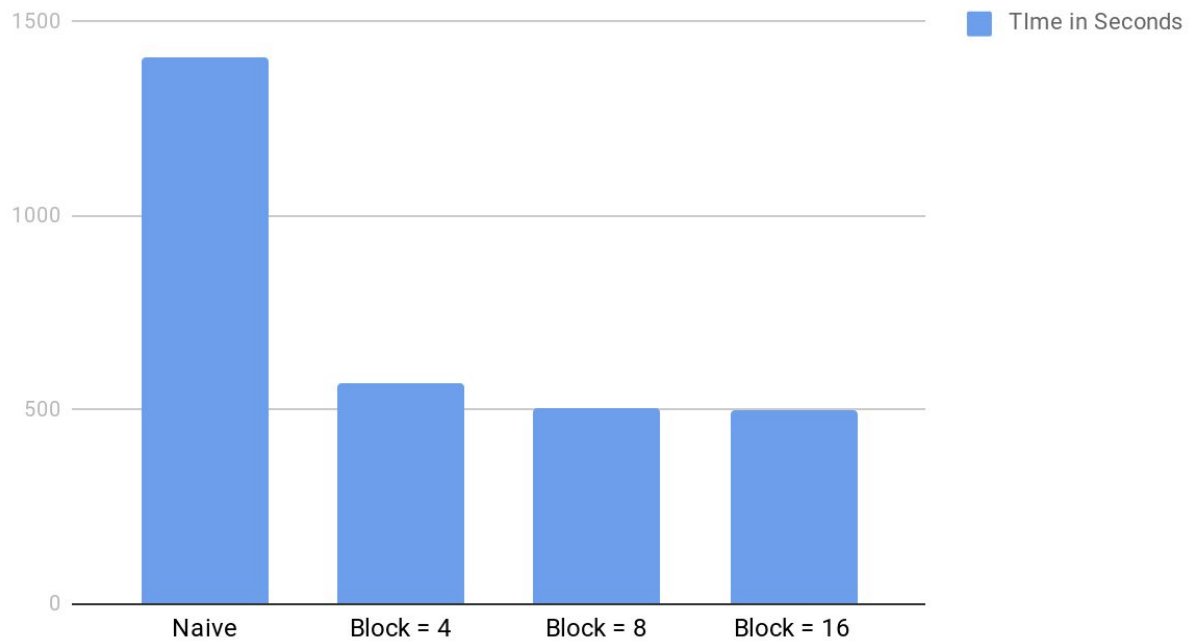
Case 2 : Block Size = 16

Number of Runs = 2

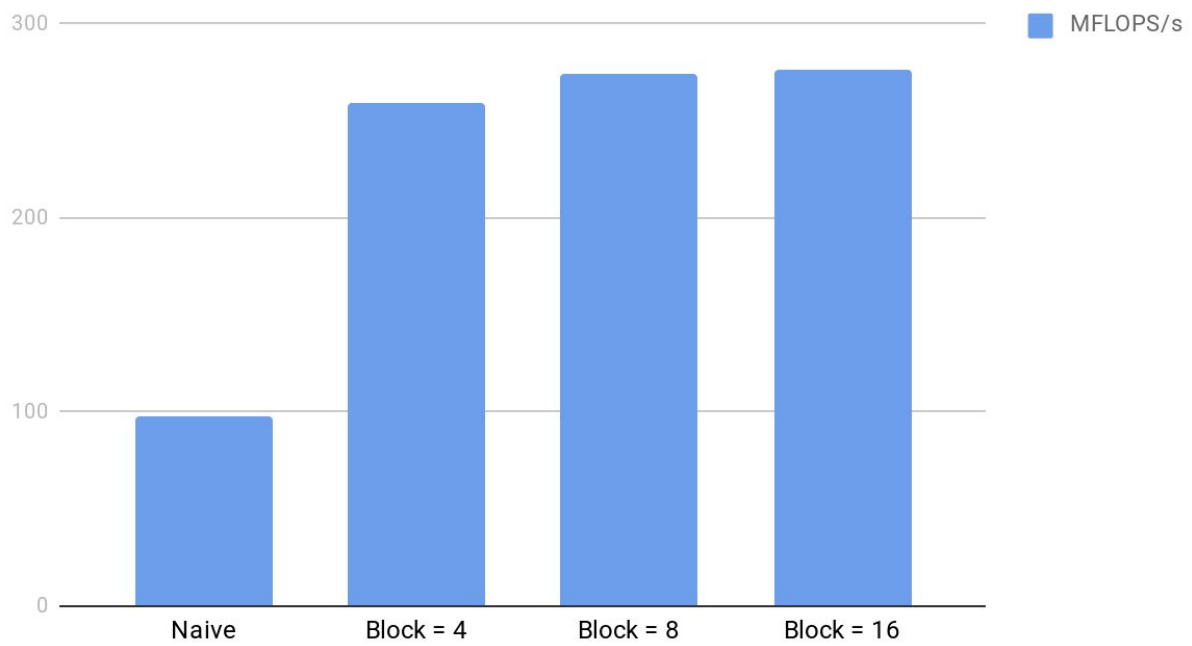
Average execution time = $(496 + 498) / 2 = 497 \text{ Seconds}$

Average performance = $(277 + 275) / 2 = 276 \text{ MFLOPS/sec}$

Execution Time



Performance



Observation:

From the plots we can understand that using blocks for matrix multiplication can improve the performance and decrease execution time. This gain is primarily due to efficient use of cache memory. We can also see that performance gain between block size = 8 and block size = 16 is low. Thus we can conclude that after a point we cannot improve performance solely by increasing block size.

```
101.775776 MFLOPs per sec
C[100][100]=879616000.000000
(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$ ./p1a
Number of FLOPs = 0, Execution time = 1371.245639 sec,
100.229273 MFLOPs per sec
C[100][100]=879616000.000000
(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$ ./p1a
Number of FLOPs = 0, Execution time = 1508.468271 sec,
91.111597 MFLOPs per sec
C[100][100]=879616000.000000
(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$ ./p1a
Number of FLOPs = 0, Execution time = 1397.306673 sec,
98.359906 MFLOPs per sec
C[100][100]=879616000.000000
(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$ ^C
(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$ ./p1b 4
Number of FLOPs = 0, Execution time = 574.948328 sec,
239.045748 MFLOPs per sec
C[100][100]=879616000.000000
(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$ ./p1b 4
Number of FLOPs = 0, Execution time = 559.247059 sec,
245.757132 MFLOPs per sec
C[100][100]=879616000.000000
(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$ ./p1b 8
Number of FLOPs = 0, Execution time = 502.354475 sec,
273.589587 MFLOPs per sec
C[100][100]=879616000.000000
(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$ ./p1b 8
Number of FLOPs = 0, Execution time = 504.142631 sec,
272.619186 MFLOPs per sec
C[100][100]=879616000.000000
(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$ ./p1b 16
Number of FLOPs = 0, Execution time = 496.090528 sec,
277.044100 MFLOPs per sec
C[100][100]=879616000.000000
(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$ ./p1b 16
Number of FLOPs = 0, Execution time = 498.778690 sec,
275.550973 MFLOPs per sec
C[100][100]=879616000.000000
(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$ ./p2
Time Taken
1.55734(base) anand@anand-Inspiron-5559:~/EE451/EE_451_F_2019_PHW_1$
```

Problem 2: Kmeans Algorithm

Execution time for 30 iterations = 1.56 sec

Output:

