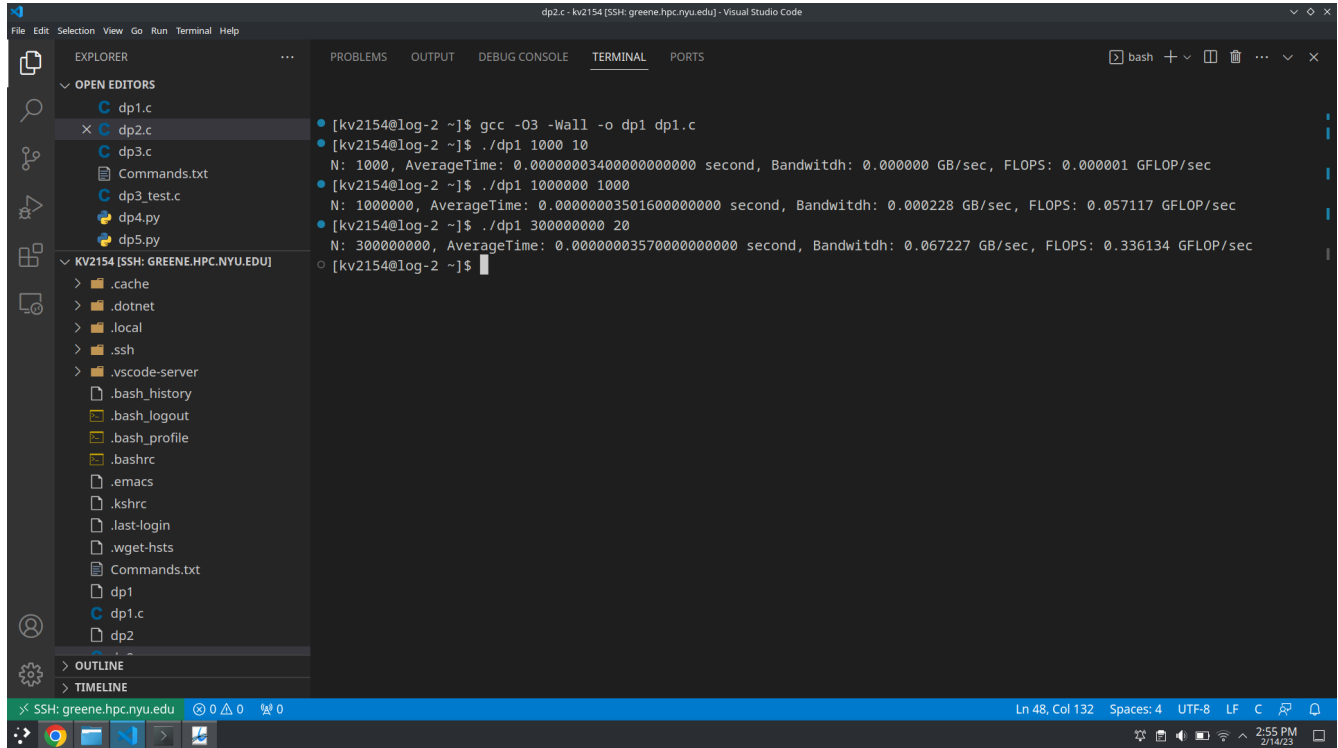


# Karan Vora (kv2154)

## Introduction to High-Performance Machine Learning (ECE-GY 9143) Assignment-1

### C1 Results



```
dp2.c - kv2154 [SSH: greene.hpc.nyu.edu] - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
dp1.c
dp2.c
dp3.c
Commands.txt
dp3_test.c
dp4.py
dp5.py
KV2154 [SSH: GREENE.HPC.NYU.EDU]
.cache
.dotnet
.local
.ssh
.vscode-server
.bash_history
.bash_logout
.bash_profile
.bashrc
.emacs
.kshrc
.last-login
.wget-hsts
Commands.txt
dp1
dp1.c
dp2
OUTLINE
TIMELINE
SSH: greene.hpc.nyu.edu 0 0 0
Ln 48, Col 132 Spaces: 4 UTF-8 LF C
2:55 PM 2/14/23

[kv2154@log-2 ~]$ gcc -O3 -Wall -o dp1 dp1.c
[kv2154@log-2 ~]$ ./dp1 1000 10
N: 1000, AverageTime: 0.000000340000000000 second, Bandwidth: 0.000000 GB/sec, FLOPS: 0.000001 GFLOP/sec
[kv2154@log-2 ~]$ ./dp1 1000000 1000
N: 1000000, AverageTime: 0.00000035016000000000 second, Bandwidth: 0.000228 GB/sec, FLOPS: 0.057117 GFLOP/sec
[kv2154@log-2 ~]$ ./dp1 300000000 20
N: 300000000, AverageTime: 0.00000035700000000000 second, Bandwidth: 0.067227 GB/sec, FLOPS: 0.336134 GFLOP/sec
[kv2154@log-2 ~]$
```

### C2 Results

```
dp1.c - kv2154 [SSH: greene.hpc.nyu.edu] - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
dp1.c
dp2.c
dp3.c
Commands.txt
dp3_test.c
dp4.py
dp5.py
KV2154 [SSH: GREENE.HPC.NYU.EDU]
.emacs
.kshrc
.last-login
.wget-hsts
Commands.txt
dp1
dp2
dp2.c
dp3
dp3_test
dp3_test.c
dp3.c
dp4.py
dp5.py
test
test.c
OUTLINE
TIMELINE
SSH: greene.hpc.nyu.edu 0 0 0 0
Ln 45, Col 58 Spaces: 4 UTF-8 LF C
[kv2154@log-2 ~]$ gcc -O3 -Wall -o dp2 dp2.c
[kv2154@log-2 ~]$ ./dp2 1000 10
N: 1000, AverageTime: 0.000000334000000000 second, Bandwidth: 0.000001 GB/sec, FLOPS: 0.000002 GFLOP/sec
[kv2154@log-2 ~]$ ./dp2 1000000 1000
N: 1000000, AverageTime: 0.000000350800000000 second, Bandwidth: 0.000912 GB/sec, FLOPS: 0.228050 GFLOP/sec
[kv2154@log-2 ~]$ ./dp2 300000000 20
N: 300000000, AverageTime: 0.000000351000000000 second, Bandwidth: 0.273504 GB/sec, FLOPS: 1.367521 GFLOP/sec
[kv2154@log-2 ~]$
```

## C3 Result

```
dp3.c - kv2154 [SSH: greene.hpc.nyu.edu] - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
dp3.c
dp1.c
Commands.txt
dp4.py
dp5.py
KV2154 [SSH: GREENE.HPC.NYU.EDU]
.cache
.dotnet
.local
.ssh
.vscode-server
.bash_history
.bash_logout
.bash_profile
.bashrc
.emacs
.kshrc
.last-login
.wget-hsts
Commands.txt
dp1
dp1.c
dp2
dp2.c
dp3
OUTLINE
TIMELINE
SSH: greene.hpc.nyu.edu 0 0 0 0
Ln 33, Col 1 Spaces: 4 UTF-8 LF C
[kv2154@log-2 ~]$ gcc -I /share/apps/intel/19.1.2/mkl/include/ -L /share/apps/intel/19.1.2/mkl/lib/intel64/ -o dp3 dp3.c -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm
[kv2154@log-2 ~]$ ./dp3 1000 10
N: 1000, AverageTime: 0.0000002416918429003 second, Bandwidth: 0.0000002416918429003 GB/sec, FLOPS: 0.0000006042296072508 GFLOP/sec
[kv2154@log-2 ~]$ ./dp3 1000000 1000
N: 1000000, AverageTime: 0.00062527046800000001 second, Bandwidth: 0.0000001279446321140 GB/sec, FLOPS: 0.00000319861580284966 GFLOP/sec
[kv2154@log-2 ~]$ ./dp3 300000000 20
N: 300000000, AverageTime: 0.02055898980000000001 second, Bandwidth: 0.00000011673725330609 GB/sec, FLOPS: 0.00000058368626653047 GFLOP/sec
[kv2154@log-2 ~]$
```

C4 Result

```
dp4.py - kv2154 [SSH: greene.hpc.nyu.edu] - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER
OPEN EDITORS
dp4.py
dp1.c
dp2.c
dp3.c
Commands.txt
dp3_test.c
KV2154 [SSH: GREENE.HPC.NYU.EDU]
.cache
.dotnet
.local
.vscodeserver
.bash_history
.bash_logout
.bash_profile
.bashrc
.emacs
.kshrc
.last-login
.wget-tests
Commands.txt
dp1
dp1.c
dp2
dp2.c
dp3
dp3_test.c
dp3.c
dp4.py
dp5.py
test.c
test.c

TERMINAL
[kv2154@log-2 ~]$ python3 dp4.py 1000 10
Average Time: 0.002951209433376789 second, Bandwidth: 0.0027758111326672847 GB/sec, FLOPS: 0.006776882648113488 GFLOP/sec
[kv2154@log-2 ~]$ python3 dp4.py 1000000 1000
^C
[1]* Stopped python3 dp4.py 1000000 1000
[kv2154@log-2 ~]$ python3 dp4.py 1000000 1
Average Time: 0.6289848029478073 second, Bandwidth: 0.0128830887140782 GB/sec, FLOPS: 0.0032286948818423856 GFLOP/sec
[kv2154@log-2 ~]$ python3 dp4.py 1000000 10
Average Time: 2.8967769583687186 second, Bandwidth: 0.0027617562950049152 GB/sec, FLOPS: 0.006904225836111422 GFLOP/sec
[kv2154@log-2 ~]$ python3 dp4.py 1000000 180
Average Time: 29.581433585770428 second, Bandwidth: 0.0002711797708657376 GB/sec, FLOPS: 0.0067793315676858154 GFLOP/sec
[kv2154@log-2 ~]$ python3 dp4.py 1000000 1000
Average Time: 317.36755874659866 second, Bandwidth: 2.5207970315541082e-05 GB/sec, FLOPS: 0.006381841334693238 GFLOP/sec
[kv2154@log-2 ~]$ python3 dp4.py 300000000 1
Average Time: 176.2971683760183 second, Bandwidth: 0.013605661654871756 GB/sec, FLOPS: 0.003401415141604728 GFLOP/sec
[kv2154@log-2 ~]$ python3 dp4.py 300000000 20
Average Time: 1834.4668369432911 second, Bandwidth: 0.0013082821360777704 GB/sec, FLOPS: 0.00654141815787004 GFLOP/sec
[kv2154@log-2 ~]$
[kv2154@log-2 ~]$
```

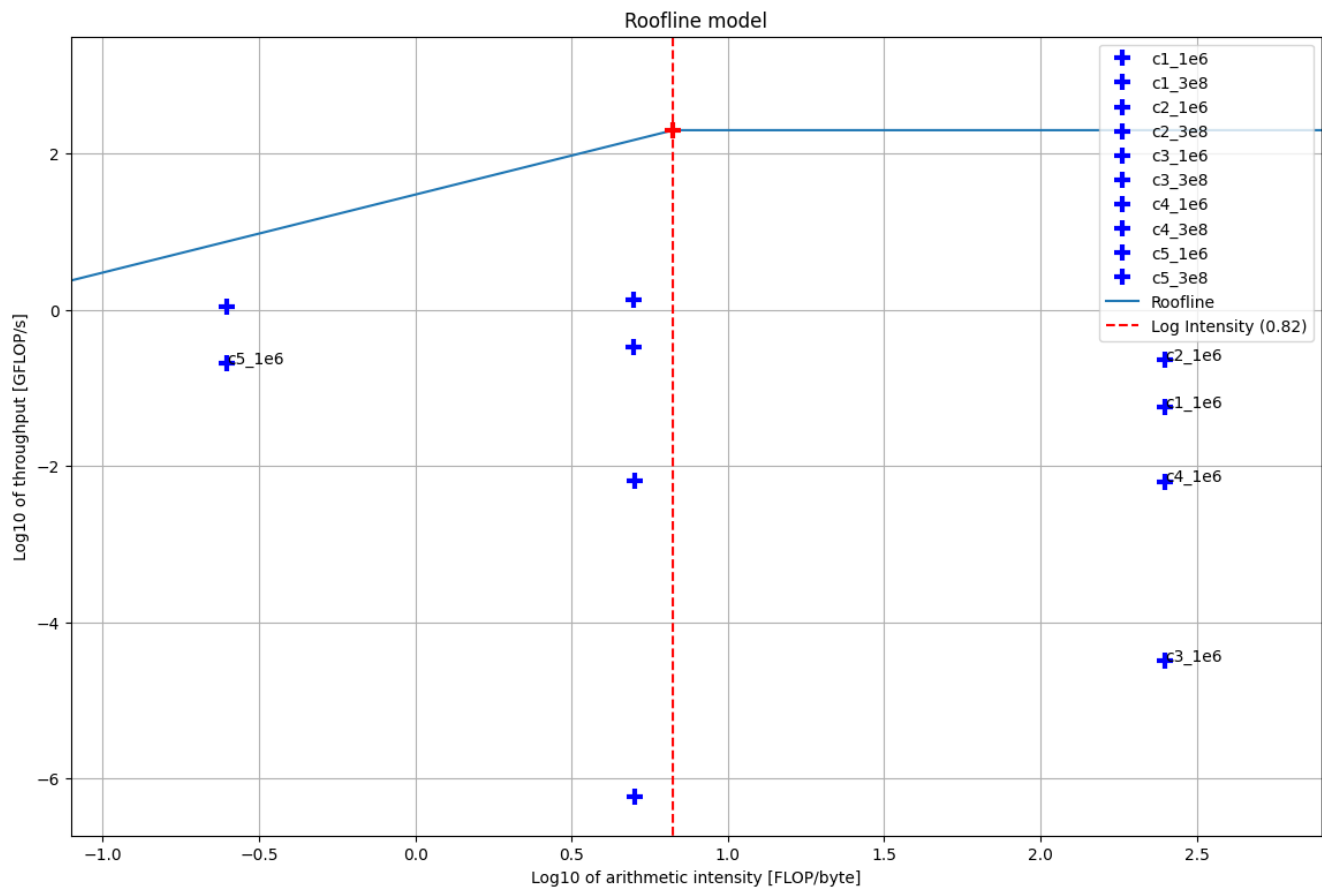
C5 Result

Q1:

If only 2<sup>nd</sup> Half of the dataset, The computed average will not be the true representation of computed dataset. Mean is a statistical summary of the whole dataset so when only a part of it is used, It wont reflect the true tendency of the dataset.

In addition, using only the second half of the measurements can lead to a biased estimate of the mean. This is because the portion of the data that is used may be systematically different from the entire dataset in some way. For example, the second half of the measurements may be systematically higher or lower than the first half. This would result in a biased estimate of the mean.

Q2:



Q3:

At  $N = 300000000$ , the average time per iteration for dp1 is 35.7 nanoseconds at bandwidth of 0.067227 GB/sec and average FLOPS is 0.336134 GFLOP/sec.

At  $N = 300000000$ , the average time per iteration for dp2 is 35.1 nanoseconds at bandwidth of 0.273504 GB/sec and average FLOPS is 1.367521 GFLOP/sec.

At  $N = 300000000$ , the average time per iteration for dp3 is 0.02055 nanoseconds at bandwidth of 0.00000011673 GB/sec and average FLOPS is 0.0000005836 GFLOP/sec.

From above mentioned data we can extrapolate that though execution time for dp1 and dp2 is similar, the flops performance and memory access is far higher for dp1 vs dp2. Meaning the arithmetic intensity for dp2 is higher than that of dp1. dp3 has the lowest performance for the same size of array, it takes the most time to perform the dot product thus the arithmetic intensity of dp3 is the lowest.

=====

Q5:

At N = 1000, the average time per iteration for dp1 is 34 nanoseconds at bandwidth of 0.000002 GB/sec and average FLOPS is 0.000001 GFLOP/sec.

At N = 1000000, the average time per iteration for dp1 is 35.01 nanoseconds at bandwidth of 0.000228 GB/sec and average FLOPS is 0.057117 GFLOP/sec.

At N = 300000000, the average time per iteration for dp1 is 35.7 nanoseconds at bandwidth of 0.067227 GB/sec and average FLOPS is 0.336134 GFLOP/sec.

At N = 1000, the average time per iteration for dp5 is 3.175 microseconds at bandwidth of 2.58 GB/sec and average FLOPS is 6.2998 GFLOP/sec.

At N = 1000000, the average time per iteration for dp5 is 0.004 seconds at bandwidth of 2.0013 GB/sec and average FLOPS is 500.323 GFLOP/sec.

At N = 300000000, the average time per iteration for dp1 is 0.46 seconds at bandwidth of 5.218319 GB/sec and average FLOPS is 26.09 GFLOP/sec.

As per above mentioned observations we can clearly see that dp1 executes much faster than dp5 even though the GLOPS performance is higher indicating that dp1 is much more efficient and the arithmetic intensity of dp1 is higher because it requires much less GFLOPS to finish the same task with same size of data. dp1 is also much more memory efficient as compared to dp1 requiring much less of resources to execute the same process.