Homework 1

ECE-GY 9143 Intro to High Performance Machine Learning

Aditya Wagh – amw9425

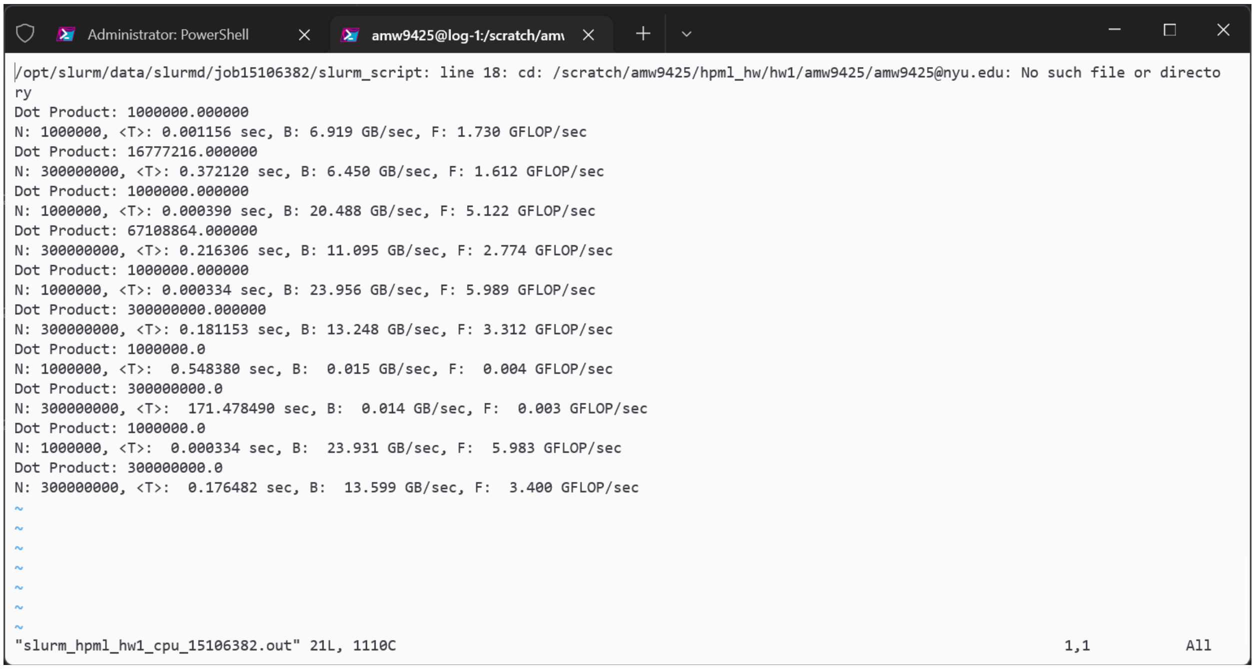
Coding Questions

Results

Program	Array Size	Iterations	Time Taken(sec)	Bandwidth (GB/sec)	FLOPS(GFLOP/sec)
Dp1.c	1,000,000	1000	0.001156	6.919	1.730
Dp1.c	300,000,000	20	0.372120	6.450	1.612
Dp2.c	1,000,000	1000	0.000390	20.488	5.122
Dp2.c	300,000,000	20	0.216306	11.095	2.774
Dp3.c	1,000,000	1000	0.000334	23.956	5.989
Dp3.c	300,000,000	20	0.181153	13.248	3.312
Dp4.py	1,000,000	1000	0.548380	0.0015	0.004
Dp4.py	300,000,000	20	171.478490	0.0014	0.003
Dp5.py	1,000,000	1000	0.000334	23.931	5.983
Dp5.py	300,000,000	20	0.176482	13.599	3.400

Screenshots

The order in this screenshot is the same as the one mentioned in the above table.



```
/opt/slurm/data/slurmd/job15106382/slurm_script: line 18: cd: /scratch/amw9425/hpml_hw/hw1/amw9425/amw9425@nyu.edu: No such file or directo
ry
Dot Product: 1000000.000000
N: 1000000, <T>: 0.001156 sec, B: 6.919 GB/sec, F: 1.730 GFLOP/sec
Dot Product: 16777216.000000
N: 300000000, <T>: 0.372120 sec, B: 6.450 GB/sec, F: 1.612 GFLOP/sec
Dot Product: 1000000.000000
N: 1000000, <T>: 0.000390 sec, B: 20.488 GB/sec, F: 5.122 GFLOP/sec
Dot Product: 67108864.000000
N: 300000000, <T>: 0.216306 sec, B: 11.095 GB/sec, F: 2.774 GFLOP/sec
Dot Product: 1000000.000000
N: 1000000, <T>: 0.000334 sec, B: 23.956 GB/sec, F: 5.989 GFLOP/sec
Dot Product: 300000000.000000
N: 300000000, <T>: 0.181153 sec, B: 13.248 GB/sec, F: 3.312 GFLOP/sec
Dot Product: 1000000.0
N: 1000000, <T>:  0.548380 sec, B:  0.015 GB/sec, F:  0.004 GFLOP/sec
Dot Product: 300000000.0
N: 300000000, <T>:  171.478490 sec, B:  0.014 GB/sec, F:  0.003 GFLOP/sec
Dot Product: 1000000.0
N: 1000000, <T>:  0.000334 sec, B:  23.931 GB/sec, F:  5.983 GFLOP/sec
Dot Product: 300000000.0
N: 300000000, <T>:  0.176482 sec, B:  13.599 GB/sec, F:  3.400 GFLOP/sec
~
~
~
~
~
~
~
"slurm_hpml_hw1_cpu_15106382.out" 21L, 1110C                                                                    1,1           All
```

Theory Questions

Q1

Using second half of the measurements ensures that we don’t consider the overhead for cache misses, and extra time it takes to fetch from memory as a result of cache misses. We can say that, as a result, our code is much more stable during the last half of the iterations.

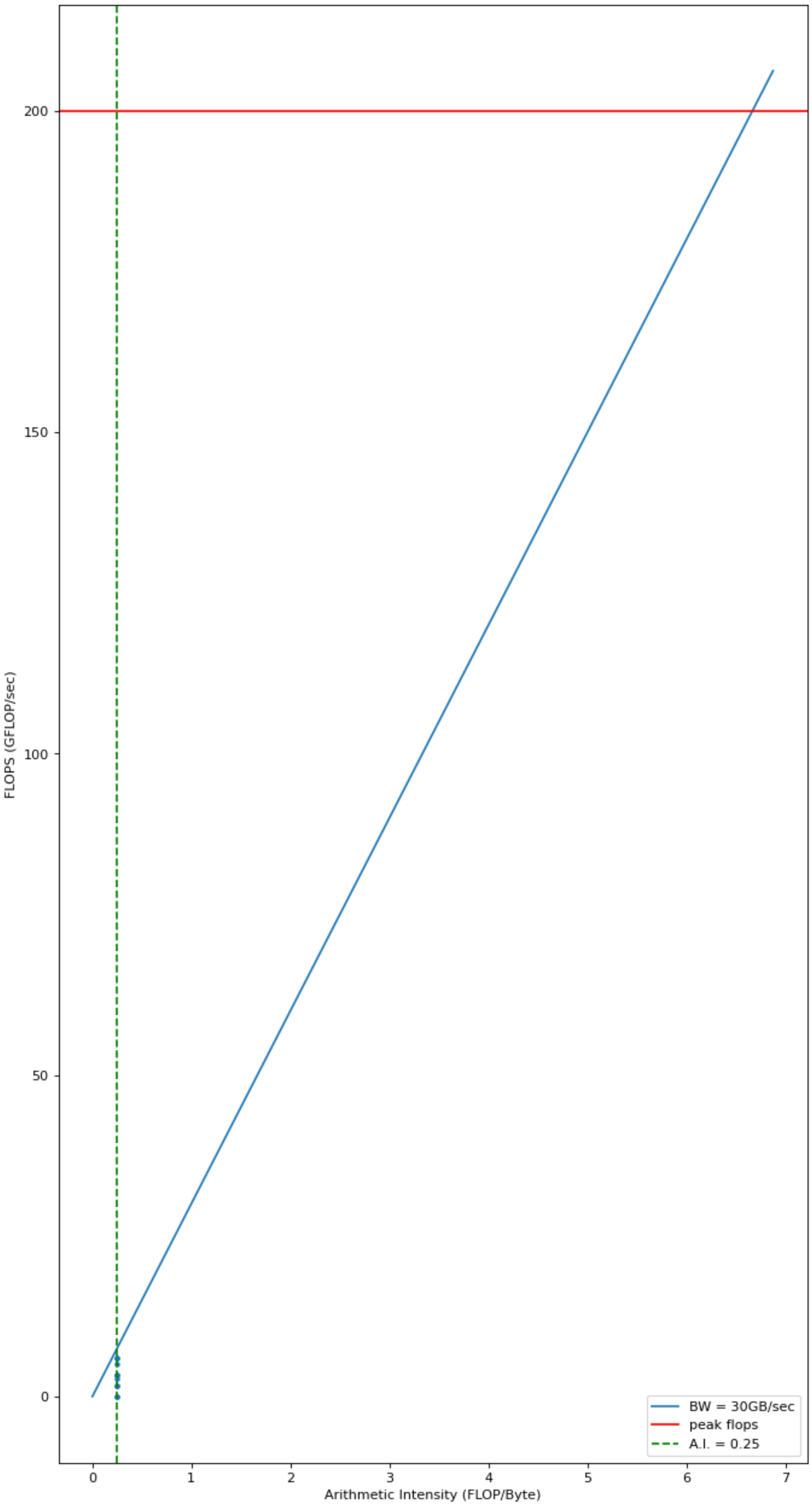
Q2 See the next page. Moved to fit image in the page.

Q3

The dp1.c program is slower since we are computing less operations per iteration of the for loop. The dp1.c program runs for whole N iterations too! Whereas the programs dp2.c and dp3.c unroll the for loop and run for N/4 and N/5 times respectively, while doing more operations per iteration of the loop. More information about **cblas_sdot** can be found out at http://www.netlib.org/lapack/explore-html/d0/d16/sdot_8f_source.html#l00081. This is a definition of **sdot** function, which is a FORTRAN function called by the **cblas_sdot** function. We can see that it unrolls the loop into 5 terms (similar to 4 terms in dp4) when the stride is set to 1.

Q4

Floating point integers are not exact integers. We are multiplying the numbers 300,000,000 times, which is very high. Small errors, thus add up and cause huge deviations in computations. The error might also be due to the fact that the cache probably gets filled up beyond 100,000 size arrays. We don’t see computational errors at 100,000 sized arrays. So errors in computations might be a result of that too!



200
150
100
50
0
FLOPS (GFLOP/sec)
0
1
2
3
4
5
6
7
Arithmetic Intensity (FLOP/Byte)
BW = 30GB/sec
peak flops
A.I. = 0.25

References

[1] https://github.com/RRZE-HPC/kerncraft/blob/master/kerncraft/roofline-plot.py

[2] https://www.cs.colostate.edu/~cs575dl/Sp2013/lects/Roofline.pdf

[3] https://sites.google.com/nyu.edu/nyu-hpc/training-support/tutorials/slurm-tutorial