TA Comment: Calculate the Floating Point operation in GFLOP/sec in the assignment although in the assignment it is mentioned FLOP/sec.

C1 – output

```
(base) [zz3904@log-1 a1]$ srun ./dp1 1000000 1000
srun: job 15065740 queued and waiting for resources
srun: job 15065740 has been allocated resources
dp 1
N: 1000000; <T>: 0.000000018 sec; B: 450146.298 GB/sec; F: 112536.574 GFlop/sec
```

```
(base) [zz3904@log-1 a1]$ srun --mem=6000 ./dp1 300000000 20
srun: job 15065768 queued and waiting for resources
srun: job 15065768 has been allocated resources
dp 1
N: 300000000; <T>: 0.000000019 sec; B: 125000000.000 GB/sec; F: 31250000.000 GFlop/sec
```

C2 – output

```
(base) [zz3904@log-1 a1]$ srun ./dp2 1000000 1000
srun: job 15065770 queued and waiting for resources
srun: job 15065770 has been allocated resources
dp 2
N: 1000000; <T>: 0.000000018 sec; B: 450045.005 GB/sec; F: 393789.379 GFlop/sec
```

```
(base) [zz3904@log-1 a1]$ srun --mem=6000 ./dp2 300000000 20
srun: job 15065771 queued and waiting for resources
srun: job 15065771 has been allocated resources
dp 2
N: 300000000; <T>: 0.000000020 sec; B: 122448979.592 GB/sec; F: 107142857.143 GFlop/sec
```

C3 – output

```
(/scratch/zz3904/envs_dirs/html) [zz3904@log-1 a1]$ srun ./dp3 1000000 1000
srun: job 15065797 queued and waiting for resources
srun: job 15065797 has been allocated resources
dp 3
N: 1000000; <T>: 0.000318320 sec; B: 25.132 GB/sec; F: 6.283 GFlop/sec
```

```
(/scratch/zz3904/envs_dirs/html) [zz3904@log-1 a1]$ srun --mem=6000 ./dp3 300000000 20
srun: job 15065798 queued and waiting for resources
srun: job 15065798 has been allocated resources
dp 3
N: 300000000; <T>: 0.191937218 sec; B: 12.504 GB/sec; F: 3.126 GFlop/sec
```

C4 – output

```
(/scratch/zz3904/envs_dirs/html) [zz3904@log-1 a1]$ srun python dp4.py 10000 100
srun: job 15068758 queued and waiting for resources
srun: job 15068758 has been allocated resources
dp 4
N: 10000;<T>: 0.005878941740000001 sec; B: 0.013640550212358457 GB/sec; F: 0.0034019728183256326 GFLOP/sec
```

```
(/scratch/zz3904/envs_dirs/html) [zz3904@log-1 a1]$ srun python dp4.py 300000 20
srun: job 15068761 queued and waiting for resources
srun: job 15068761 has been allocated resources
dp 4
N: 300000;<T>: 0.17536481 sec; B: 0.013686850856793904 GB/sec; F: 0.0034214389990785497 GFLOP/sec
```

C5 - output

```
(/scratch/zz3904/envs_dirs/html) [zz3904@log-1 a1]$ srun python dp5.py 10000 1000
srun: job 15068774 queued and waiting for resources
srun: job 15068774 has been allocated resources
dp 5
N: 10000;<T>: 2.9433640000000003e-06 sec; B: 27.245016246716343 GB/sec; F: 6.794946190821115 GFLOP/sec
```

```
(/scratch/zz3904/envs_dirs/html) [zz3904@log-1 a1]$ srun python dp5.py 300000 20
srun: job 15068775 queued and waiting for resources
srun: job 15068775 has been allocated resources
dp 5
N: 300000;<T>: 9.81915e-05 sec; B: 24.443989551030384 GB/sec; F: 6.11050854707383 GFLOP/sec
```

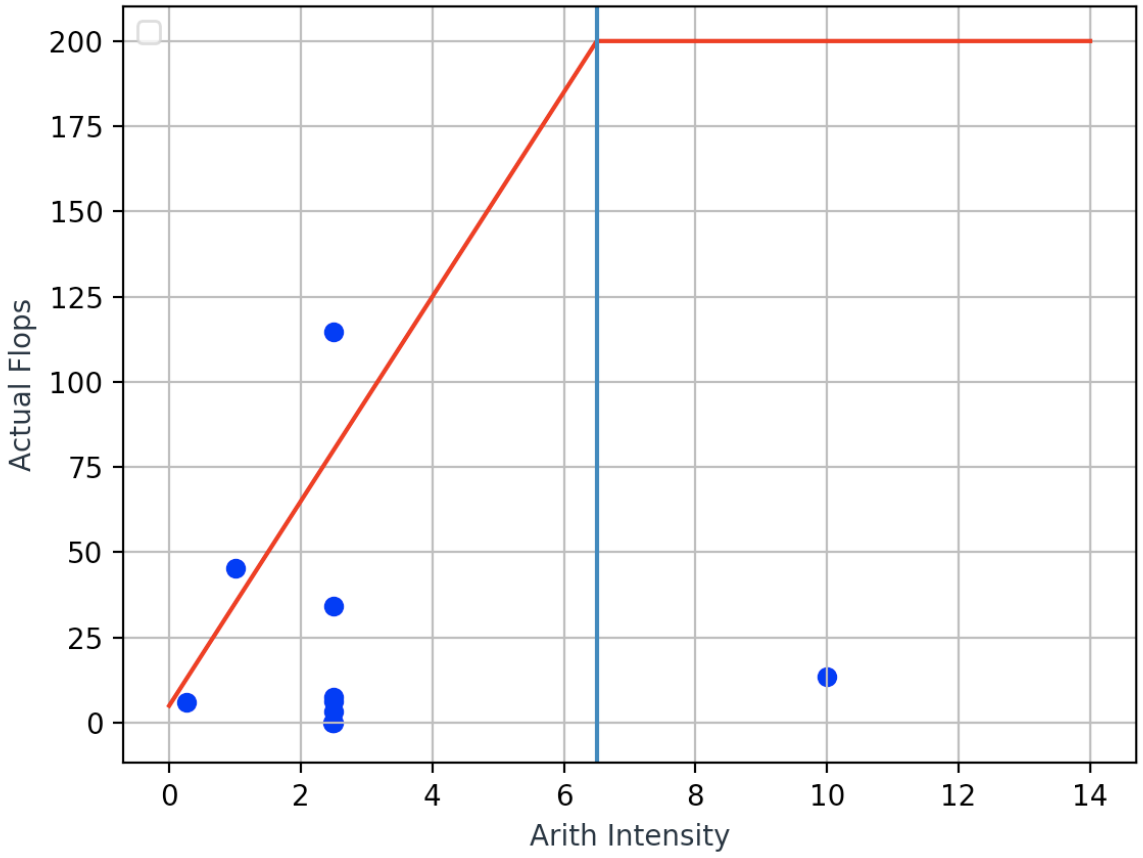
Q1 (3 points):

Explain the consequence of only using the second half of the measurements for the computation of the mean.

- using the second half of the measurements for the computation of the mean does not correctly reflect the execution time.
- Second half is faster
- Reason: more will be cached from disk in memory which speeds up the execution.

Q2

- I wrote a python script to plot the numbers. I first copied the results from c1-c5 to make a list and plot all the points.



Q3 (5 points):
Using the N=300000000 simple loop as the baseline, explain the
the difference in performance for the other 5 measurements in the C variants.
(TA comments: In Q3 it is written to compare the baseline (which C1) with 5 measurements in the C variants. It is a mistake you need to compare with other 2 (c2 and c3) measurements.)

Answer

- Performance of c3 is better because it uses vectorization.
- Throughput of c2 better than c1 because c2 conducts more float point operations each iteration.

Q3 (6 points):
Check the result of the dot product computations against the analytically calculated result. Explain your findings. (Hint: Floating point operations are not exact.)

Answer

- When using N=300000000, dot product results from c1 and c2 are different from the analytically calculated result, because floating-point decimal values generally do not have an exact binary representation.