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

**C1 – output**

**Text

Description automatically generated**

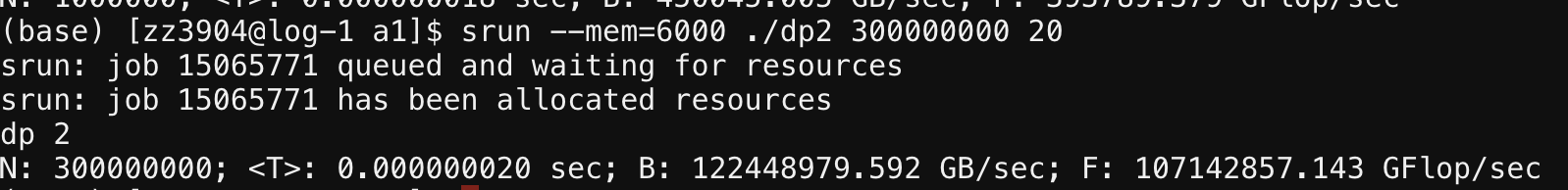
**Text

Description automatically generated**

**C2 – output**

**Text

Description automatically generated**

****

**C3 – output**

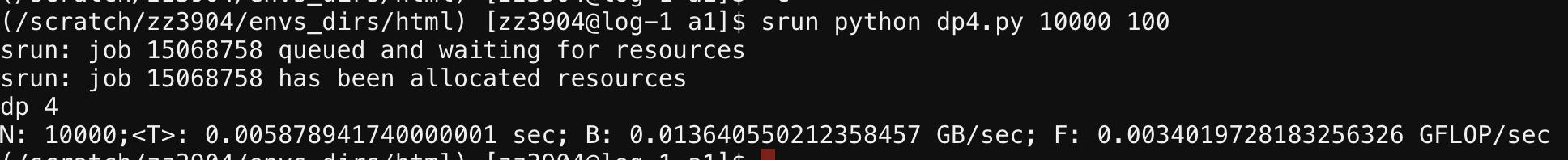
**Text

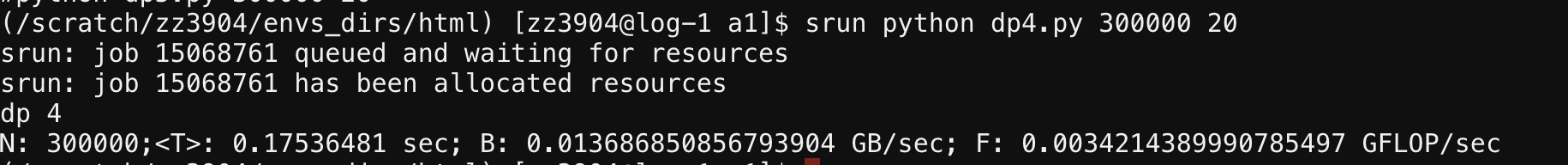
Description automatically generated**

**Text

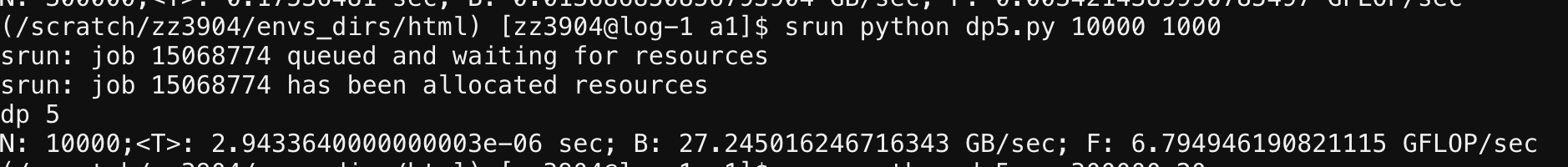
Description automatically generated**

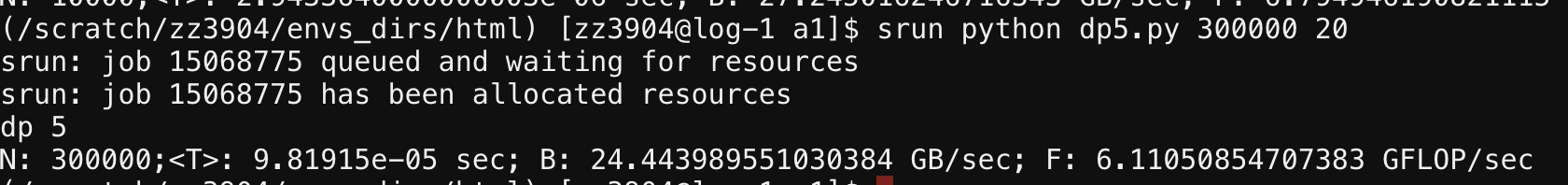
**C4 – output**

****

****

**C5 - output**

****

****

**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.

**Chart, line chart, scatter chart

Description automatically generated**

**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.