David Vondran

1/23/2022

CS450

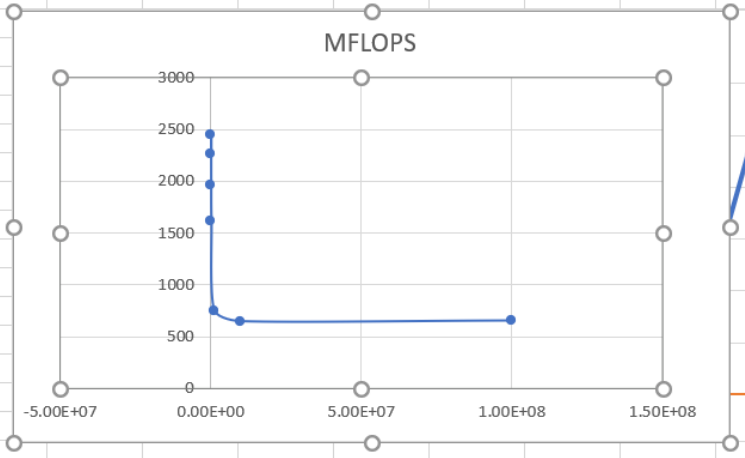
HW1 – Report

During this assignment, I ran into significant problems attempting to understand the goals and requirements of the assignments, which massively reduced the amount of time that I was able to actually perform correct work. Overall, the goals and correct results were achieved, but many of the graphs are poor looking or ill-representative of the data.

Running task 1 on the data given, I was able to replicate the results given in the textbook correctly. I Used the values of N that were given in the textbook (10e1, 10e2, 10e3, 10e4, 10e5, 10e6, 10e7), and was able to receive the results listed below.

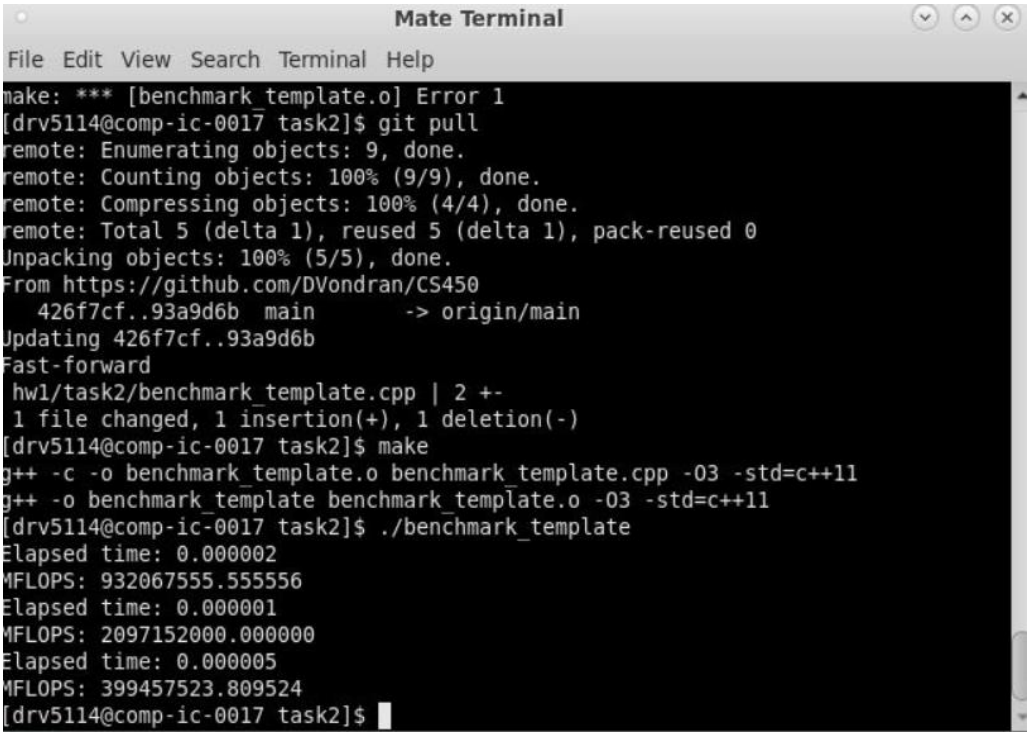
```
MFLOPS: 399457523.809524
[drv5114@comp-ic-0017 task2]$ cd ..
[drv5114@comp-ic-0017 hw1]$ cd task1
[drv5114@comp-ic-0017 task1]$ ./benchmark_template
Elapsed time: 0.088381
MFLOPS: 2262.934588
Elapsed time: 0.081697
MFLOPS: 2448.070694
Elapsed time: 0.101877
MFLOPS: 1963.152143
Elapsed time: 0.123473
MFLOPS: 1619.785126
Elapsed time: 0.265033
MFLOPS: 754.622990
Elapsed time: 0.307779
MFLOPS: 649.816758
Elapsed time: 0.304784
MFLOPS: 656.202297
[drv5114@comp-ic-0017 task1]$
```

N	MFLOPS
1.00E+02	2262.936
1.00E+03	2448.077
1.00E+04	1963.152
1.00E+05	1619.785
1.00E+06	754.623
1.00E+07	649.8168
1.00E+08	656.2023



The values themselves format in the correct shape of the given graph, but excel makes the graph look bad due to the exponential growth of the x axis. I did not have time to fix this, so I hope you can excuse said graph and focus predominantly on the values given.

Task 2 was more easy to visualize, with the 3 different states being tested in 3 separate runs. The first state was where C contained all positive values of 1, state 2 was where C contained all negative values of -1, and state 3 had random values between -1 and 1.



The results of this can be seen at the bottom of this screenshot, where state 1 of all positive values had the second largest value, state 2 had the largest, and state 3 with the random values was the slowest. I expected the negative and positive valued arrays to have roughly the same time, while the alternating values would be the slowest, however the positive values seemed to lag behind significantly. I am not sure why this is the case, however, it does make sense that the alternating would be slower than the same values as the system would have to determine whether to add or subtract the values.

Note, I included the .pbs files in the final submission, as the path to the source code was changed.