Recap

When I processed 2GB data (i.e. data generated at Scale-Factor = 2), I got some unexpected results, in which the runtime for 4 threaded operation was more than that of 1 threaded operation -

```
aryan@SpaceShip:~/DRIVES/CS/Zattabolt/tpch-query5/build$ time ./tpch_query5 --r_name ASIA --st
art_date 1994-01-01 --end_date 1995-01-01 --threads 1 --table_path ../data --result_path singl
e-thread-output.txt
TPCH Query 5 implementation completed.

real    18m2.210s
user    5m41.597s
sys     2m22.538s
```

1 Thread: 18.2 min

```
aryan@SpaceShip:~/DRIVES/CS/Zattabolt/tpch-query5/build$ time ./tpch_query5 --r_name ASIA --st
art_date 1994-01-01 --end_date 1995-01-01 --threads 4 --table_path ../data --result_path four-
thread-output.txt
TPCH Query 5 implementation completed.

real    24m4.341s
user    6m27.667s
sys     3m15.073s
```

4 Thread: 24.4 min

Experimentation

As 2GB was a big chunk of data, i experimented with smaller sized data, like 100MB, 500MB and 1GB:

```
aryan@SpaceShip:~/DRIVES/CS/Zattabolt/TRIAL AND ERR
OR/tpch-query5_submission/build$ time ./tpch_query5
 --r_name ASIA --start_date 1994-01-01 --end_date 1
995-01-01 --threads 1 --table_path ../data/small --
result_path single-thread-output-halfgb_5.txt



TPCH Query 5 implementation completed.

real    0m6.371s
user    0m5.913s
sys     0m0.451s
```

100 MB, 1 Thread: 0.637 min

```
aryan@SpaceShip:~/DRIVES/CS/Zattabolt/TRIAL AND ERR
OR/tpch-query5_submission/build$ time ./tpch_query5
 --r_name ASIA --start_date 1994-01-01 --end_date 1
995-01-01 --threads 1 --table_path ../data/halfgb -
-result_path single-thread-output-halfgb_4.txt

TPCH Query 5 implementation completed.

real    1m17.583s
user    1m12.482s
sys     0m5.008s
```

500 MB, 1 Thread: 1.175 min

```
aryan@SpaceShip:~/DRIVES/CS/Zattabolt/TRIAL AND
 ERROR/tpch-query5_submission/build$ time ./tpc
h_query5 --r_name ASIA --start_date 1994-01-01
--end_date 1995-01-01 --threads 1 --table_path
../data/onegb --result_path singleThread-output
-onegb.txt
TPCH Query 5 implementation completed.

real    8m54.411s
user    2m48.673s
sys     1m2.402s
```

1 GB, 1 Thread: 8.544 min

```
aryan@SpaceShip:~/DRIVES/CS/Zattabolt/TRIAL AND ERR
OR/tpch-query5_submission/build$ time ./tpch_query5
 --r_name ASIA --start_date 1994-01-01 --end_date 1
995-01-01 --threads 4 --table_path ../data/small --
result_path single-thread-output-halfgb_5.txt



TPCH Query 5 implementation completed.

real    0m6.239s
user    0m6.100s
sys     0m0.442s
```

100 MB, 4 Threads: 0.623 min

```
aryan@SpaceShip:~/DRIVES/CS/Zattabolt/TRIAL AND ERR
OR/tpch-query5_submission/build$ time ./tpch_query5
 --r_name ASIA --start_date 1994-01-01 --end_date 1
995-01-01 --threads 4 --table_path ../data/halfgb -
-result_path four-thread-output-halfgb_5.txt

TPCH Query 5 implementation completed.

real    1m14.691s
user    1m14.911s
sys     0m5.026s
```

500 MB, 4 Threads: 1.147 min

```
aryan@SpaceShip:~/DRIVES/CS/Zattabolt/TRIAL AND
 ERROR/tpch-query5_submission/build$ time ./tpc
h_query5 --r_name ASIA --start_date 1994-01-01
--end_date 1995-01-01 --threads 4 --table_path
../data/onegb --result_path fourThread-output-o
negb.txt
TPCH Query 5 implementation completed.

real    9m2.351s
user    3m2.029s
sys     1m18.045s
```

1 GB, 4 Threads: 9.235 min

Runtime: 4 threads < 1 thread

Runtime: 4 threads < 1 thread

Runtime: 1 thread < 4 threads

Observation

- I observed that whenever I was running an operation that used RAM only, we got the expected output: 4 threads were faster than 1 thread

- But, whenever my system used the Swap memory along with RAM, the results were opposite, with 1 thread being faster than 4 threads.
  - Operations using RAM only:
    - 100 MB operation
    - 500 MB operation
  - Operations using RAM + Swap:
    - 1 GB operation (8 GB RAM + 8 GB swap)
    - 2 GB operation (8 GB RAM + 20 GB swap).

Checking the observation

- For ensuring my observations, I performed processing on another system, with following specs: 16 GB RAM, i5 10th Gen Processor
- (For reference, my earlier system specs: 8 GB RAM, i5 10th Gen processor)

```
anant_rs@Drowned-Dragon:~/Desktop/project/build$ time ./tpch_query5 --r_name ASIA --start_date 19
94-01-01 --end_date 1995-01-01 --threads 1 --table_path ../data/onegb --result_path one-output-on
egb.txt
TPCH Query 5 implementation completed.

real    1m38.445s
user    1m31.590s
sys     0m6.792s
```

1 Thread:
1.384 min

```
anant_rs@Drowned-Dragon:~/Desktop/project/build$ time ./tpch_query5 --r_name ASIA --start_date 19
94-01-01 --end_date 1995-01-01 --threads 4 --table_path ../data/onegb --result_path four-output-o
negb.txt
TPCH Query 5 implementation completed.

real    1m28.634s
user    1m29.803s
sys     0m6.474s
```

4 Threads:
1.286 min

Observation with 16 GB RAM system:
We get the desired result: **Runtime: 4 threads < 1 thread**

Conclusion

- When the program uses only RAM, we get the expected result — the 4-threaded version is faster than the single-threaded one due to parallel processing.
- However, when the system starts using a mix of RAM and Swap memory (because of hardware limitations), we observe the opposite: the single-threaded version performs better.
- The likely reason is an **I/O bottleneck**. In the 4-threaded version, multiple threads try to access data at the same time. But Swap memory (which uses the hard drive) is much slower than RAM. This leads to heavy disk I/O, causing threads to wait and slowing down the overall execution.
- So, instead of gaining speed through multithreading, the program suffers due to the slower disk access — a classic example of an I/O bottleneck.

NOTE: In the new system, I checked for only scale-factor 1 data because it requires almost 16 GB RAM, and processing data any bigger than scale-factor 2 would have rendered the same issue as my system.