

Mahek Aggarwal, Vivian Li

DS 4300

February 5th, 2024

Twitter Performance with Redis

In this assignment, we utilized Redis, a key-value database to store posted tweets to simulate users retrieving their home timelines and posting tweets on Twitter. The hardware configurations included an Apple MacBook Pro with a M3 processor chip with 16GB of RAM and 8 cores.

Our results:

Number of tweets processed in one second:	8677.82
Home timelines retrieved per second:	21609.80

Compared to our results from last time, when we used MySQL as our database, we have been able to achieve faster home timeline retrievals and a greater number of tweets processed per second. This is due to the fact that a non-relational database stores its data in memory and excels at “reading” the data faster than SQL databases. However, some further possible improvements for the future include optimizing the way that the key-value pairs in the database correspond to each other to ensure faster reading and writing to the database. Pipeline and batch operations can also be used to send multiple operations into the function simultaneously to reduce the number of trips between the client and the server. We can also expire the key for older tweets so that only the recents and most relevant ones will be used for implementations.

Both team members contributed equally to this project as we read and understood the assignment together. We discussed the installation and use of Redis and how to implement it for this assignment, as it is a new tool that neither of us has used before. We modified our functions from last time and both wrote some of the functions for the assignment