Ethan Gilworth
Rithesh Kayathi

Report On Redis Performance in a Mock Twitter

Strategy 1:

Posting tweets was very fast because it only required a simple set operation in Redis. The key-value store was designed for quick data retrieval and storage, which is why the tweet posting was fast. However, constructing the home timeline on the fly required looking up the tweets of each user being followed, which was a more time-consuming process. Overall performance in this strategy was roughly 7500 posts per second and roughly 620 retrievals. This does make it a strictly dominating strategy on the initial relational database implementation, but between this strategy and strategy 2 you do have a toss up. Perhaps we could have made more optimizations to make the timeline retrieval even faster without costing ourselves too much posting speed.

Strategy 2:

In this strategy, each tweet was automatically copied (or a reference to the tweet) to the user's home timeline as it was posted. The write performance was slower due to the additional step of copying the tweet to the user's home timeline. We used sorted sets when posting to timelines in hopes of making this operation very fast. Unfortunately we were only able to retrieve about 2000 timelines per second which is middle of the pack relative to other students. This might have to do with the fact that sorted sets are slow to index and retrieve. I am unsure. We were only able to post about 3500 tweets per second because of all the overhead that went into appending a posted tweet to all appropriate timelines. This slow posting time potentially makes this a less viable strategy than even the SQL implementation, though the pathetically slow timeline retrieval in that strategy does make it quite unappealing.

Conclusion:

It can be concluded that both of these strategies are likely preferable to a relational database implementation. Picking between these two strategies, though, is quite difficult. I think I would personally air on the side of strategy 1 as posting tweets quickly seems like the top priority to me, and it still retrieves timelines relatively quickly. Strategy 2 could be the preferred option if your priority is timeline retrieval and are okay with slow tweets posting.