CS2309 Assignment 2

Tan Yee Jian (A0190190L)

September 7, 2021

1 Problem Statement

Log-structured Merge Trees (LSM Trees) have a good write throughput, but high read amplification. This is the opposite of the commonly used update-in-place storage systems which have great read performance, but low write throughput. As observed at Yahoo!, the low latency workloads that emphasize on writes are increasing from 10-20% to about 50%, which is getting unsuitable for update-in-place systems. LSM Trees are a good candidate for write-intensive tasks, but modifications need to be made so it has a better real-world performance.

1.1 Solution

Therefore, the researchers have come up with bLSM, a LSM Tree with the advantages of B-Trees. This is done by adding Bloom filters in LSM Trees (hence the *b* in *bLSM*) to increase index performance and replacing the merge scheduler in LSM Trees with the new "spring and gear" merge scheduler. When measured under a new performance metric, *read fanout* that is claimed to better characterize real-world index operations, *bLSM* outperform B-Trees in most cases.

2 Pros and Cons

2.1 Pros of *bLSM*

1. Using Bloom filter reduces read amplification by trimming the search tree, from N to $1 + \frac{N}{100}$. This allows *bLSM* to have near-optimal read performance.

2. The new "spring and gear" scheduler solves the problem of *snowshoveling*, so application writes are not blocked for long period of times, improving write latencies.

2.2 Cons