Paper Evaluation: Optimizing Space Amplification in RocksDB

Michael Lanthier

RocksDB is a persistent key value store that optimizes space efficiency over throughput and latency. By exploiting the append only nature and compactness of LSM trees RocksDB is able to keep the data on each node extremely compact unlike the B+ trees used in MySql/InnoDb which maintain 50-75% vacancy. In addition to the use of LSM trees, data compression algorithms increase compactness with a small detriment to read performance. Despite its goals for data storage efficiency, RocksDB was able to increase transaction throughput with only a slight decrease in read performance.

Questions/Comments

1. RocksDB seems to be able to maintain good enough read/write performance because the nodes do not see a ton of traffic because of their sharding technique. Would RocksDB see a large decrease in performance if these nodes began to see more churn in the data? With their compression techniques it seems like large amounts of reads and writes at a nodewould result in more CPU time being spent compressing and uncompressing data. While the goal of RocksDB is not to achieve extreme speeds and high throughput, would it reasonably be able to handle that throughput and is it good enough to be used outside of Facebooks use case?
2. Are cascading compactions an expensive operation to complete? Would large amount sof updates and insertions result in more cascading compactions that could impact database performance or would it just be negligible.