CSE 6350 Advanced Topics in Computer Architecture

Lab 02 - WiscKey Implementation

Ezequiel Aguilar Gonzalez
1001096975

Introduction

Key-value databases, a kind of NoSQL databases, is data storage system for persistent storage and retrieving key-value pairs using unique keys. Most key-value stores use **LSM-tree** [1] for indexing key-value pairs. **LevelDB** [4], is one of the widely-used key-value stores designed by Google. **WiscKey** [2] shows that, as the LSM-tree size grows, write and read amplification are getting bigger due to LevelDB's compaction operation. This drastically decreases the performance of LSM-tree based key-value stores and reduces device lifetime.

To deal this issue, Wisckey's central idea is to separate keys and values. First, only keys are kept sorted in the LSM-tree, while values are stored separately in a log. As a result, key sorting and garbage collection are decoupled. Final, as values are unsorted, parallel random-reads are used for range queries to exploit SSDs (Solid State Drives) internal parallelism.

In this document, we focus on to implement WiscKey on top of LevelDB and compare the performance of our system with original LevelDB. This design separates keys from values to minimize Input/Output amplification.

Design

Our WiscKey implementation is storing the value in value log file according to the written order and save only its offset in the file and length information in the record.

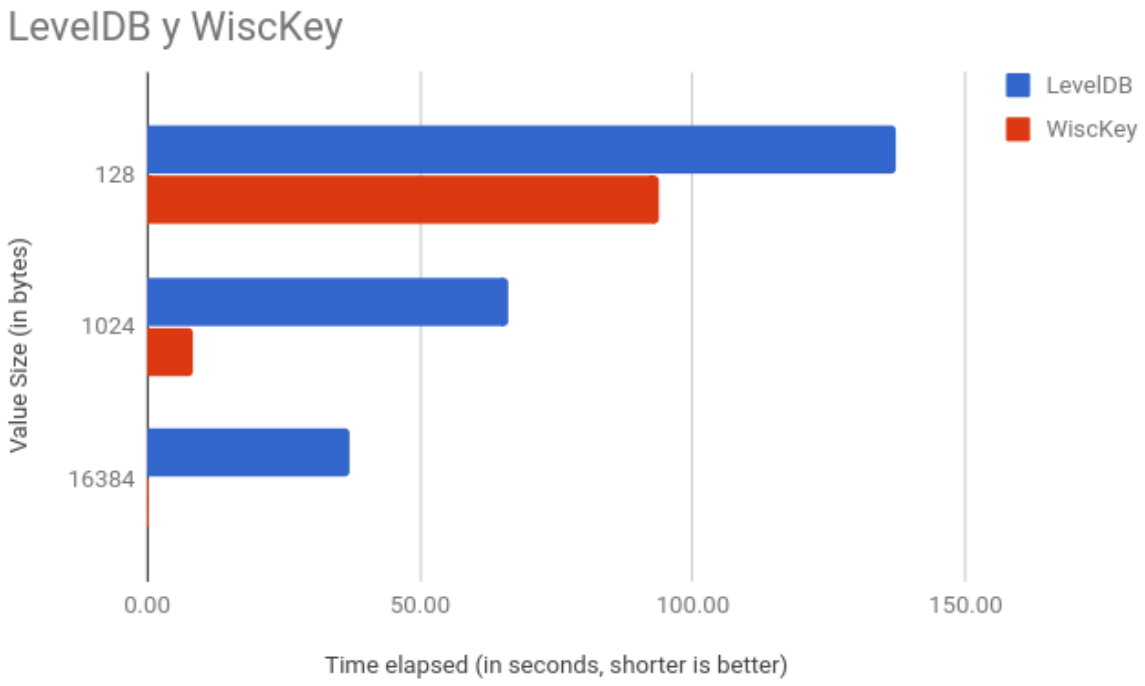
Setup

We rented a virtual machine from Google Cloud, which provides 60GB storage, 2 virtual cores along with 4GB RAM.

Test by Benchmark

Value Size	LevelDB (seconds)	WiscKey (seconds)
128	137.08	93.65
1024	66.00	8.27
16384	37.22	0.40

The next figure show us the performance between LevelDB and WisKey with value size of 128 bytes, 1024 bytes and 16384 bytes.



Future work

SSDs have made gains in random I/O performance. One interesting research would be to mix the value-log approach [2], and keep value and keys pointers in the B+-tree [3].

References

[1] O'NEIL, P., CHENG, E., GAWLICK, D., AND O'NEIL, E. The log-structured merge-tree (LSM-tree). Acta Inf. 33, 4 (1996), 351–385.

[2] Lanyue Lu, Thanumalayan Sankaranarayana Pillai, Hariharan Gopalakrishnan, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau. 2017. WiscKey: Separating Keys from Values in SSD-Conscious Storage. ACM Trans. Storage 13, 1, Article 5 (March 2017), 28 pages.

[3] J. S. Ahn, C. Seo, R. Mayuram, R. Yaseen, J. S. Kim and S. Maeng, "ForestDB: A Fast Key-Value Storage System for Variable-Length String Keys," in IEEE Transactions on Computers, vol. 65, no. 3, pp. 902-915, March 1 2016.

[4] https://github.com/google/leveldb