Team Members: Simran Sandhu, Rahul Sai, Shulin Pan, Eva Liu

Title: Exploring the Evolution of Storage Devices and its Impact on the Performance of Key-Value Stores

Introduction:

With the improvements in storage-media and CPUs in the past years, the benchmarks of popular databases have shown varying results in terms of latency, R/W throughput, both due to implementation details and the underlying hardware. The bottlenecks have shifted from the storage media to the CPUs and NICs. For our project, the primary objective is to explore this shift in bottleneck, especially on flash-based storage interfaces such as SATA and NVMe, by running various types of workloads on popular KV stores. The secondary objective is to compare the R/W throughput and latency of newer write-optimized KV stores (LSM/Bϵ-Tree), and older B+tree based KV stores. We also intend to explore how additional IO capabilities (PCIe lanes) [1] provided by server-grade CPUs can affect the utilization when compared to consumer-grade CPUs.

Background/Motivation:

Storage devices have come a long way, from disk-based storage to the now widely used Solid State Drives (SSDs), which have begun to be widely used since late 2000s for their efficient data accesses using NAND-based flash memory. NVMe is optimized for the low-latency, high-bandwidth characteristics of SSDs. Since the mid-2010s, NVMe SSDs have become more affordable and have started to replace SATA SSDs in many applications, especially in high-performance computing and gaming. Since the advent of newer storage interfaces, the bottlenecks have shifted more towards CPU and networks.

RocksDB [2] and SplinterDB [3] are write-optimized key value stores, with RocksDB (LSM-Based) optimized for flash-based storage and SplinterDB (Bϵ-Tree based) optimized for NVMe respectively. SplinterDB outperforms RocksDB on NVMe SSDs in terms of throughput and latency according to YCSB benchmarks. *YCSB [4] is a widely-used tool for benchmarking cloud-based databases, using performance and scaling tiers to simulate real-world application workloads.

High Level Design:

1. Collate core YCSB workloads into a text file.
2. Traverse the text file and execute the operations on a combination of SSD Types and KV Stores (WiredTiger, RocksDB, SplinterDB).
3. Measure the latency and R/W throughput.
4. Measure the hardware utilization (CPU and Disk) using Linux utils like iostat.
5. Use grep and regex to filter out the required details from the results.
6. Run a background script which combines steps 3 to 5 and logs these details in the form of a CSV file.
7. Use the CSV file to generate graphs and visualize the various aspects of the experiment.
8. Test out this setup locally on SATA and NVMe drives, and then proceed to CloudLab [5] for the actual experiment.

Intended Results:

- We are trying to reproduce the result of the SplinterDB [3] paper where SplinterDB outperforms RocksDB significantly on NVMe SSDs. We are also trying to explore if this behavior remains unaffected on SATA based SSDs.
- We will also compare the hardware utilization of Write optimized DBs on various type of SSDs i.e., SATA and NVMe, to analyze how the evolution of storage devices and interfaces have shifted the bottleneck from disks to CPUs.

References:

[1] T. Fountain, A. McCarthy, F. Peng, and others, “PCI express: An overview of PCI express, cabled PCI express and PXI express,” in *10th ICALEPCS Int. Conf. on Accelerator & Large Expt. Physics Control Systems*, 2005.

[2] F. Yang, K. Dou, S. Chen, M. Hou, J.-U. Kang, and S. Cho, “Optimizing NoSQL DB on Flash: A Case Study of RocksDB,” in *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, 2015, pp. 1062–1069. doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.197.

[3] A. Conway *et al.*, “SplinterDB: Closing the Bandwidth Gap for NVMe Key-Value Stores,” in *Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference*, 2020.

[4] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, “Benchmarking Cloud Serving Systems with YCSB,” in *Proceedings of the 1st ACM Symposium on Cloud Computing*, 2010, pp. 143–154. doi: 10.1145/1807128.1807152.

[5] R. Ricci, E. Eide, and C. Team, “Introducing CloudLab: Scientific infrastructure for advancing cloud architectures and applications,” *the magazine of USENIX & SAGE*, vol. 39, no. 6, pp. 36–38, 2014.