Boosting Compaction Performance of LSM--tree-based KV Stores in Multi-Near-Data Processing Systems

Hui Sun, Qiang Wang
Anhui University
sunhui@ahu.edu.cn, e19201078@stu.ahu.edu.cn

Yinliang Yue, Yuhong Zhao
Chinese Academy of Sciences
yueyinliang@iie.ac.cn, zhaoyuhong@iie.ac.cn

Song Fu
University of North Texas
Song.Fu@unt.edu

Problem

The LSM--tree has been main data structure of KV stores, which comprises one memory component and multiple disk components.

- Compaction may lead to serious write amplification which is a performance bottleneck. That is when random writes become intensive, the compaction time is too long, during which user-level write/read operations are blocked.
- In addition, the compaction process also leads to many resource costs, such as data movement and CPU cycles.

MStore Overview

Figure 1 shows the overview system of MStore prototype, including four modules.

- NDP-cluster management module
- NDP-cluster storage balancing module
- NDP-cluster computing load balancing module.
- Data transmission protocol and the semantic interface between the master and slave systems.

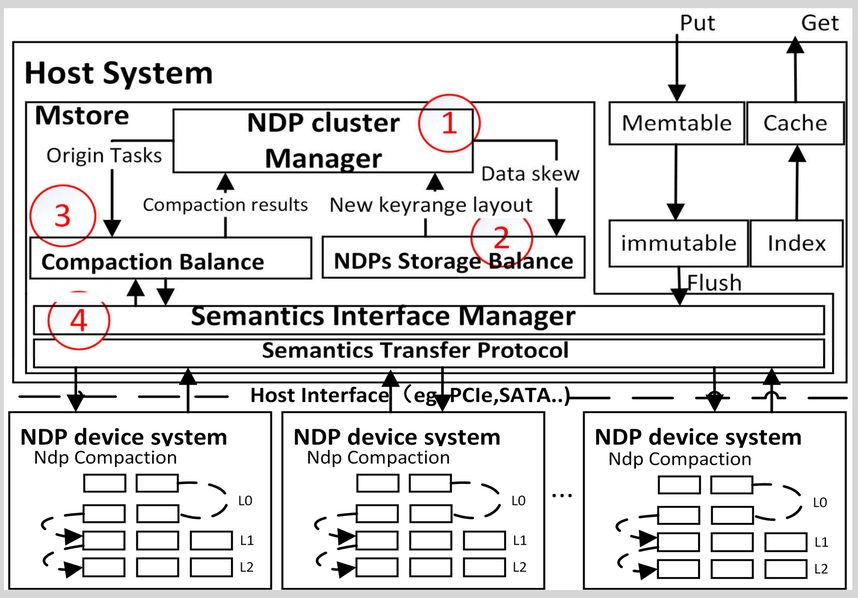

Figure 1 Overview of MStore

Fundamental Technologies

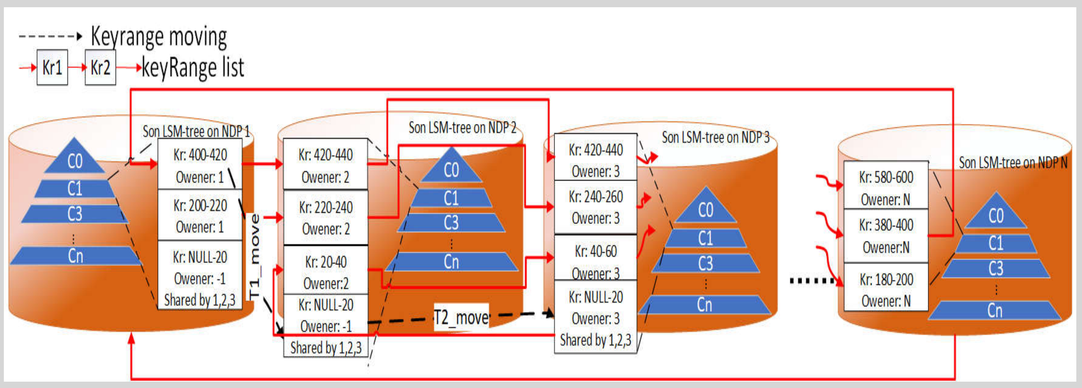

Figure 2: Data Layout In MStore

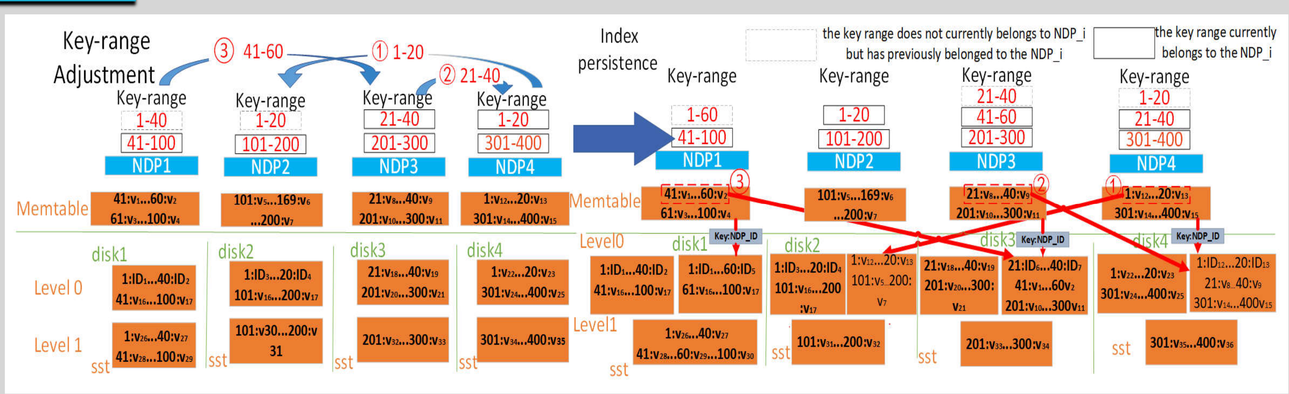

Figure 3: Data Layout In MStore

Multi-column LSM--tree divides the dataset into multiple key ranges, and each device manages the data of one or more key ranges. Longitudinal slicing of the LSM-Tree has the following benefits:

- Each device performs compaction on its own managed data.
- The key ranges of each device can be dynamically adjusted to avoid a data skew in the system.
- In addition, the data in the same column is highly independent; thus, data movement is avoided during compaction tasks on NDP devices.

MStore controls the load of each device by adjusting the key range to which each device belongs. This procedure includes two steps:

- Key-range adjustment. This procedure needs to achieve two goals since we do not perform data migration, we need to achieve two goals after adjustment. First is avoiding data storage skew and second is that the impact on read operations is minimized, that is, the overlap area in the system LSM--tree is minimized. So we propose a key range priority-based adjustment algorithm.
- Index persistence of overlapping area. After key-range adjustment, the host distributes corresponding data to each NDP according to the new key ranges. An NDP needs to insert <key:NDP_ID> in the previously owned key range (the key range represented by the dotted box in Figure 2) These key ranges contain expired data; and then, the <key:NDP_ID> will overwrite these expired data in the compaction.

In the synchronous mode, the workload of each NDP compaction may vary significantly; thus, extra task balancing is needed to make the workload of each NDP roughly the same, avoiding the idling of the NDP device for a long period of time and the waste of the computations and IO resources in the NDP cluster.

Results

Figure 3 and 4 show the performance of MStore comparing TStore and MStore-basic.

- MStore has 3.88 times and 3.5 times faster than TStore under DB_bench and YCSB-C workloads, respectively
- Compared with MStore-basic, MStore's load optimization strategy contributes 14\% optimization on average.
- Device-side WA of MStore reduces by 31.5% on average compared with TStore and Host-side write amplification of MStore maintained at 2.

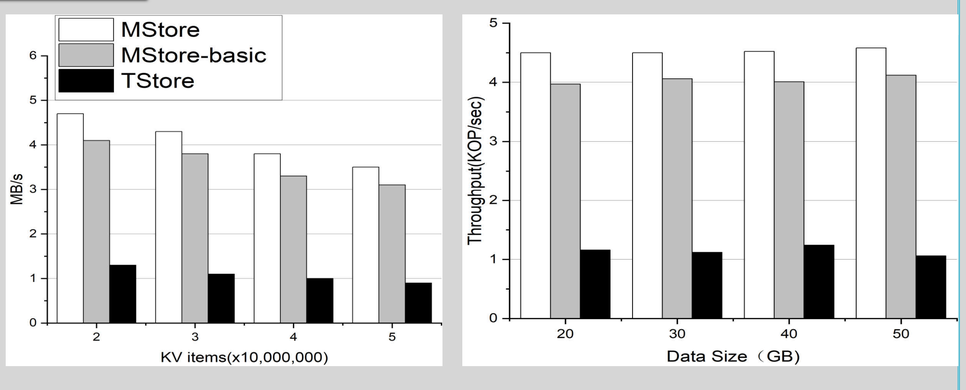

Figure 4 Throughput under DB_bench and YCSB-C

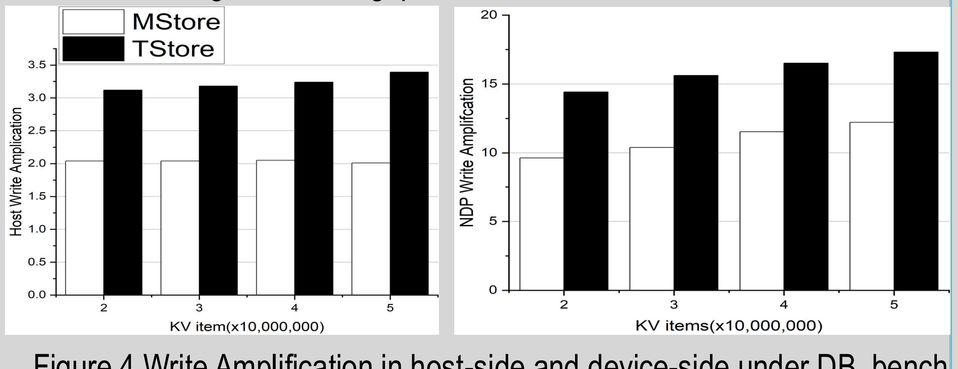

Figure 4 Write Amplification in host-side and device-side under DB_bench

Conclusion

We propose a near-data computing cluster named MStore to improve the write performance of KV stores. According to the data organization, we design a key-range dynamic adjustment-based storage balancing and compaction tasks-aware computing balancing methods to alleviate the 'cannikin law' caused by uneven compaction on each device owing to data skew issue in the NDP cluster. We also test the performance of MStore comparing TStore and MStore-basic on a real-world platform.

Poster template by ResearchPosters.co.za