A Storage Structure to Ensure High Throughput for Both Reads and Writes

Xiaodong Zhang
The Ohio State University, USA

Abstract

B+tree was designed and implemented in 1970s to store data in a sequential format, which provides fast reads. However the write throughput is low with this method due to writing data in scattered mode. To solve this problem, LSM-tree (Log-Structured Merge Tree) was developed in 1990s and implemented in Google's Big Table system 10 years after. Now LSM-tree is a standard storage format for big data systems. A distinguished merit of LSM-tree is to maximize the write throughput in a batch mode. However, for workloads mixed with both reads and writes, read performance is degraded by LSM-tree-induced cache invalidations. This problem was not in the scope of its design in 1990s because memory caching for storage data was not a common practice then.

In this talk, I will present a new storage structure called LS**b**M-tree (The Log-Structured **Buffered** Merge Tree), which retains all the merits of LSM-tree, but also addresses LSM-tree induced caching problem. We aim to ensure a high throughput for both reads and writes, and show its effectiveness in the design and in experiments. LSbM-tree is being tested and deployed in data management systems, including Cassandra and LevelDB.

About the speaker:

Xiaodong Zhang is the Robert M. Critchfield Professor in Engineering and Chair of the Computer Science and Engineering Department at the Ohio State University. His research interests focus on data management in computer and distributed systems. He has made strong efforts to transfer his academic research into advanced technology to advance the design and implementation of major general-purpose computing systems. He received his Ph.D. in Computer Science from University of Colorado at Boulder, where he received Distinguished Engineering Alumni Award in 2011. He is a Fellow of the ACM, and a Fellow of the IEEE.