World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:11, No:11, 2017

Improve B-Tree Index's Performance Using Lock-Free Hash Table

Authors : Zhanfeng Ma, Zhiping Xiong, Hu Yin, Zhengwei She, Aditya P. Gurajada, Tianlun Chen, Ying Li

Abstract : Many RDBMS vendors use B-tree index to achieve high performance for point queries and range queries, and some of them also employ hash index to further enhance the performance as hash table is more efficient for point queries. However, there are extra overheads to maintain a separate hash index, for example, hash mapping for all data records must always be maintained, which results in more memory space consumption; locking, logging and other mechanisms are needed to guarantee ACID, which affects the concurrency and scalability of the system. To relieve the overheads, Hash Cached B-tree (HCB) index is proposed in this paper, which consists of a standard disk-based B-tree index and an additional in-memory lock-free hash table. Initially, only the B-tree index is constructed for all data records, the hash table is built on the fly based on runtime workload, only data records accessed by point queries are indexed using hash table, this helps reduce the memory footprint. Changes to hash table are done using compare-and-swap (CAS) without performing locking and logging, this helps improve the concurrency and avoid contention. The hash table is also optimized to be cache conscious. HCB index is implemented in SAP ASE database, compared with the standard B-tree index, early experiments and customer adoptions show significant performance improvement. This paper provides an overview of the design of HCB index and reports the experimental results.

Keywords : B-tree, compare-and-swap, lock-free hash table, point queries, range queries, SAP ASE database
Conference Title : ICDE 2017 : International Conference on Data Engineering
Conference Location : Dubai, United Arab Emirates
Conference Dates : November 24-25, 2017