Literature Review

Cheng Liu

2016-11-17

1 Accelerating the Index Traversals for In-Memory Databases

This work [1] developed an hardware accelerator for hash index lookup which is a critical operation for database query. The accelerator was built on top of a RISC processor architecture. It shares the cache hierarchy and closely coupled with a host processor. Here are the highlights of this accelerator.

- The index operation is divided into three steps i.e. hash, walk traversal and result output. Particularly, hash and walk traversal are decoupled using a set of queues. Eventually, the two operations are pipelined.
- This work allows multiple index operations performed in parallel on the same hash table. Moreover, as the parallel walk traversals typically occurs on different rows of the hash table. These keys can fit in the same cache line thanks to the hash logic, which is also a key factor for the final good performance.
- The accelerator is built based on a RISC processor architecture. There are a few interesting design optimizations here.

  The touch instruction is developed to pre-fetch the data blocks to reduce the long memory access overhead.

  Input buffers and output buffers are nicely integrated in the processor pipeline to decouple the hash and walk traversal operations.

  A few three-operand operations are added to aid the hash operations.

Although the authors claimed that this work handles the pointer chasing based node traversal during the indexing, no optimization is actually done for this. When the node traversal of an entry of a hash table starts, it sequentially does the chasing. The system achieves the performance mainly through the multiple parallel traversals and balanced hashing.

There are few design options that may be further optimized.

- The host processor will be idled when the accelerator starts to work.

- The host processor does the index work when there are exceptions such as TLB miss. It seems that accelerator will be used only when the data set is in the main memory.
- As the cache line is typically determined, the number of the parallel keys and the size of the keys will be limited when fitting the multiple parallel keys in the same cache line.

References

[1] Onur Kocberber, Boris Grot, Javier Picorel, Babak Falsafi, Kevin Lim, and Parthasarathy Ranganathan. Meet the walkers: Accelerating index traversals for in-memory databases. In *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 468–479. ACM, 2013.