Towards Cold/Hot Range Partitioning for Compute/Memory Load Balancing in Processing-in-Memory

Takato Hideshima
hideshima-takato182@g.ecc.u-tokyo.ac.jp
The University of Tokyo
Japan

Tomoharu Ugawa
tugawa@acm.org
The University of Tokyo
Japan

Memory bus bandwidth has lagged behind CPU instruction throughput in recent years, creating a bottleneck known as the "memory wall" problem. Processing-in-memory (PIM) architectures solve this problem by placing a processing unit close to each of the hundreds or thousands of memory arrays, allowing parallel access to data without having to go through the memory bus. Applications where memory accesses are frequent and throughput is important, such as database indexes [1], are expected to benefit.

UPMEM [3], currently the only publicly available PIM device, works like a distributed memory computer in that the in-memory processors do not share the memory view. In addition, each memory array is small and each in-memory processor is weak. Therefore, to achieve high performance, it is necessary to balance the computational load among the processors while keeping the data that each of them handles small enough to fit into each memory array. In this presentation, we refer to this property as *compute/memory load balancing*.

However, it is difficult to build an ordered database index that achieves compute/memory load balancing and efficiently handles both point queries and range scans. We focus on batch queries, where a large number of queries are proceesed in batches. For range scans that retrieve contiguous entries, range-based partitioning [2] is preferred, where the range assigned to each processor is a large chunk in the search key space. The problem is that, if the partitioning only attempts to balance the number of queries, the number of entries stored in a single memory array can increase without an upper limit, and if the entries are evenly distributed, the compute load will not balance when the workload is skewed.

In this study, we propose to improve range-based partitioning by creating two types of ranges: cold/hot ranges. First, the search key space is partitioned to "cold ranges" so that each partition has the same number of entries. Next, the computational skewness is addressed by specially treating only the query-heavy intervals. Specifically, small number of query-heavy ranges are extracted as "hot ranges" so that each range has small number of entries and almost same number of queries. At this time, the number of hot ranges is kept below the number of in-memory processors, so that each processor is responsible for at most one hot range.

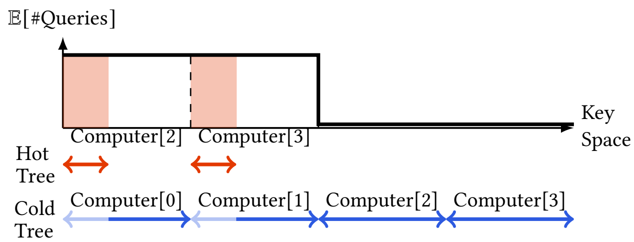

Figure 1. Cold/hot ranges.

We propose a greedy algorithm that determines how to extract hot ranges according to the query distributon. We also proved that the proposed algorithm achieves the load balancing where each in-memory processor processes

$$(1+\alpha)\frac{Q}{P} + \max\{\#\text{queries to one element}\}$$

queries and holds

$$(1+\alpha^{-1})\frac{E}{P}$$

entries, where Q is the number of queries, E is the number of entries, P is the number of in-memory processors, and $\alpha \in \mathbb{N}$ is a tunable parameter.

References

[1] Hongbo Kang, Yiwei Zhao, Guy E. Blelloch, Laxman Dhulipala, Yan Gu, Charles McGuffey, and Phillip B. Gibbons. 2022. PIM-Tree: A Skew-Resistant Index for Processing-in-Memory. *Proceedings of the VLDB Endowment* 16, 4 (Dec. 2022), 946–958. https://doi.org/10.14778/3574245.3574275

[2] Ioannis Konstantinou, Dimitrios Tsoumakos, and Nectarios Koziris. 2011. Fast and Cost-Effective Online Load-Balancing in Distributed Range-Queriable Systems. *IEEE Transactions on Parallel and Distributed Systems* 22, 8 (Aug. 2011), 1350–1364. https://doi.org/10.1109/tpds.2010.200

[3] UPMEM SAS. [n. d.]. UPMEM – UPMEM is releasing a true Processing-in-Memory (PIM) acceleration solution. https://www.upmem.com/. Accessed on 2024/7/25.

APSys '24, September 4–5, 2024, Kyoto, Japan
2024.