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 speed up.

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, the number of processors is very large and each memory array is small. 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*.

Our goal is compute/memory load balancing for range-queryable database indexes. We focus on throughput for batch queries, where a large number of queries are processed in batches. For range queries, which retrieve contiguous items, range-based partitioning [2] is preferable, where the assignment of the search key space to each processor is done in large chunks. This also results in high throughput for point queries when the workload is evenly distributed [1].

The problem arises when the workload is skewed. If the partitioning only attempts to balance the number of queries, the number of items stored in a single memory array can grow indefinitely. If the items are evenly distributed, the compute load will not balance.

We solve this problem by creating two types of ranges: cold/hot ranges (Figure 1). First, we partition the search key space into "cold ranges" so that each partition contains the same number of items. Then, we address the workload skew by giving special treatment to "hot" intervals that many queries fall into. Specifically, we extract a number of "hot ranges" that are less than or equal to the number of in-memory processors, so that they have a limited number

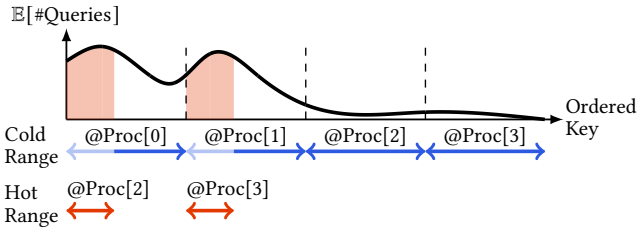

Figure 1. Cold/hot ranges.

of items and approximately the same number of queries. Finally, we assign one cold range and zero or one hot range per processor.

We propose a greedy algorithm that determines where to extract as hot ranges considering the query distribution. We also proved that the proposed algorithm achieves load balancing with each in-memory processor processing

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

queries and holding

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

items, where Q is the number of queries, E is the number of items, 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.