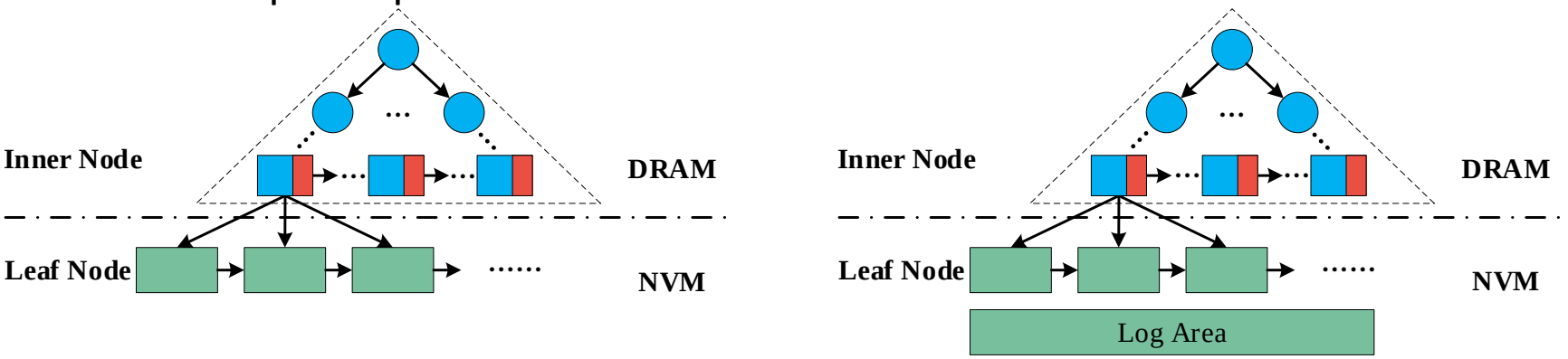
# Optimizing $B^+$-tree for Hybrid Memory with In-Node Hotspot Cache and eADR Awareness

**Peiquan JIN, Zhaole CHU, Gaocong LIU, Yongping LUO, Shouhong WAN**

Frontiers of Computer Science, DOI: 10.1007/s11704-023-3344-x

# Problems & Ideas

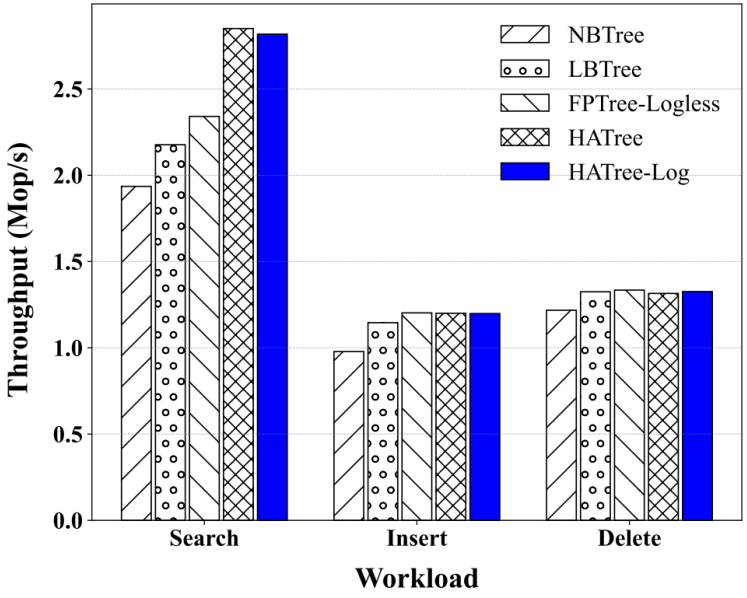
- Problems of previous $B^+$-tree for hybrid memory:
  - The design of unsorted leaf nodes avoids the costly sorting when inserting new keys, but worsen search performance.
  - The average space utilization of the inner nodes in $B^+$-tree is below 70%, i.e., there are about 30% used entries in the parent nodes of leaf nodes.
- Ideas:
  - HATree(Hotness-Aware $B^+$-tree): Utilize the unused space in the parent nodes of leaf nodes to cache hot keys.
  - HATree-Log: Use record log and eADR technology to further improve the update performance

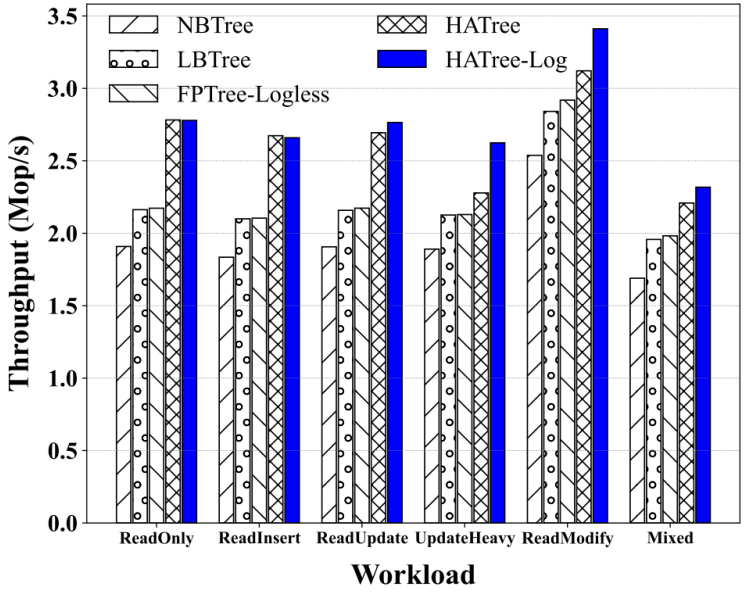



Architecture of HATree(left) and HATree-Log(right). All the inner nodes reside in DRAM and leaf nodes are placed on NVM. The rectangle colored in red is used for caching hot keys.

# Main Contributions

- Contributions:
  - We notice that the parent of leaf nodes in the $B^+$-tree contain about 30% unused space. We theoretically analyze the feasibility of using in-node hotspot caches to improve the search performance of the tree index without extra DRAM caches. And propose HATree following the idea of in-node hotspot  cache.
  - To further improve the update performance of HATree, we utilize the eADR technology to enhance HATree with instant log persistence




Experimental results of HATree and HAtree-Log. Left: The search, insert and delete performance of the five indices in the eADR mode; Right: The performance of these five indices under the YCSB workloads in the eADR mode