HM-ANN: Efficient Billion-Point Nearest Neighbor Search on Heterogeneous Memory

Jie Ren[1], Minjia Zhang[2] and Dong Li[1]

[1]University of California Merced, [2]Microsoft

Motivation

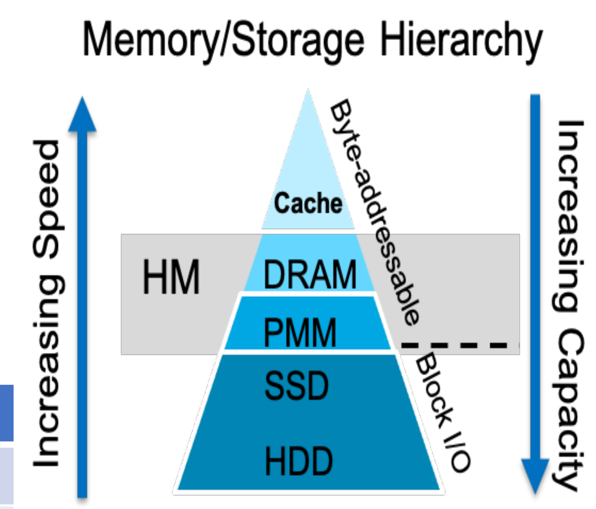
Enable fast and highly accurate billion-scale ANNS.

❑ State-of-the-art ANNS

	Graph-based ANNS	Quantization-based ANNS	SSD-based ANNS
Example	HMSW[1] NSG[2]	IMI+OPQ[3] L&C[4]	DiskANN[5]
High Recall	✓	✗	✓
Low Latency	✓	✗	✗
Memory support for billion-point	✗	✓	✓

❑ Heterogeneous Memory (HM) is Promising

- HM = fast memory + slow memory
- Fast mem (e.g., the traditional DRAM)
  - ➢ expensive ➢ fast ➢ small mem. capacity
- Slow mem (e.g., Optane PMM)
  - ➢ cheap ➢ slow ➢ large mem. capacity

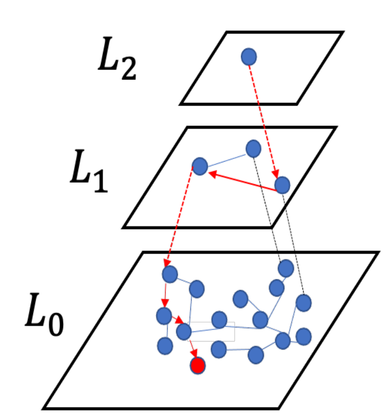


	HDD	SSD	Optane PMM	DRAM
Latency	7.1 ms	68 us	170-300 ns	100 ns
Bandwidth	2.6 MB/s	250 MB/s	39 GB/s	64 GB/s

❖ Challenge 1: The slow memory (PMM) performs ~80X times faster than SSD, but it is still 3X slower than DRAM. The existing ANNS with a naive data placement strategy can not fully enjoy the benefit of HM.

❑ Hierarchical Graph-based ANNS (HNSW)

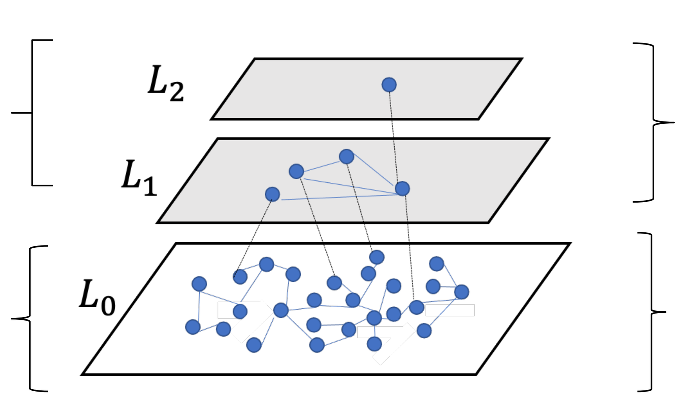
- Each layer is a navigable small world graph.
- L_0 contains all database elements, and the upper layers are randomly selected, nested subsets of database element.
- The majority of search happens in L_0 .



❖ Challenge 2: Can we take both memory and data heterogeneity into consideration, and enables billion-scale ANNS without using compression?

Overview

HM-ANN is a HM-awared graph-based ANNS, which achieves 95% top-1 recall in less than 1ms on a single CPU node.



Algorithm

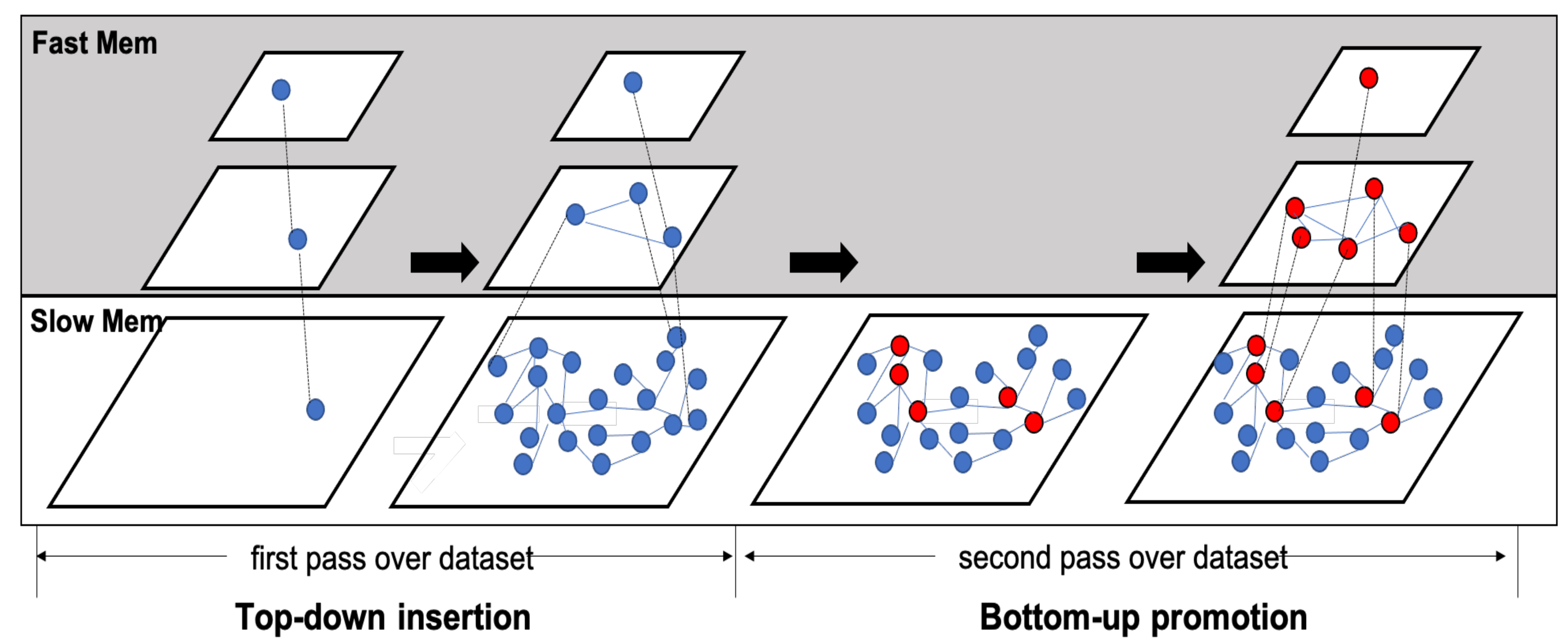
Indexing: build navigable graphs with high search quality using two passes over the dataset.

❑ Top-down insertion:

➢ Create navigable small world graph as the bottom-most layer in slow memory.

❑ Bottom-up promotion:

➢ Prioritize promoting pivot points with high degree from the bottom layer graph to form upper layers placed in fast memory.



Search: Make the majority of search happens in fast memory and minimize searches in slow memory.

❑ Upper layer search

❖ Goal: achieve high qualify search in fast memory.

➢ From the top layer to L_2 : one-greedy search

➢ In L_1 : N-greedy search; prefetch data in L_0 into fast memory.

❑ Parallel search in L_0

❖ Goal: minimize searches in slow memory while achieving high recall.

➢ In L_0 : multiple entry point with one-greedy searches in parallel.

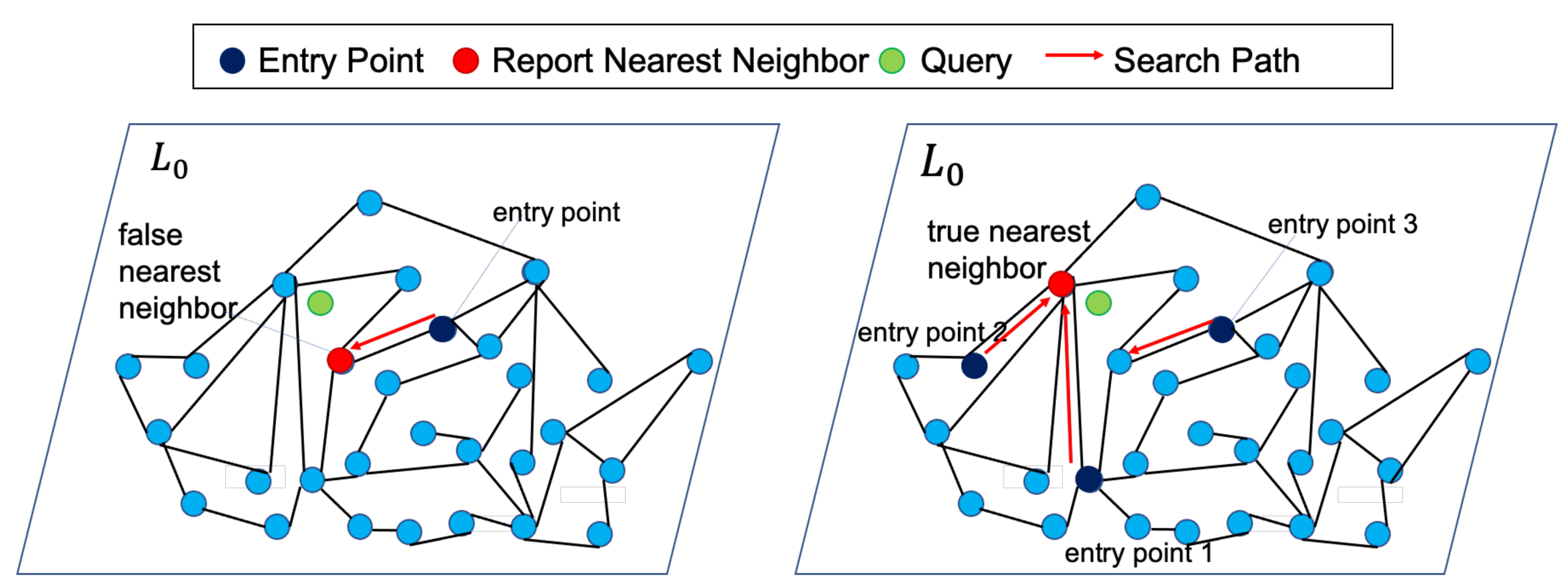


Fig. Single entry point with one-greedy search   Fig. Multiple entry points with one-greedy search

Experiment Results

HM-ANN establishes the new state-of-the-art for indexing and searching billion-point datasets.

❑ Billion-scale algorithm comparison:

➢ Testing bed

- Intel Xeon Gold 6252 CPU@2.3GHz.
- DDR4 (96GB) as fast memory and **Optane DC PMM (1.5TB)**

➢ Baselines

- We extend state-of-the-art graph-based to billion-scale ANNS (HNSW [1] and NSG[2]) on HM through using fast memory as the cache of slow memory.
- We build two state-of-the-art billion-scale quantization-based methods (IMI+OPQ[3] and L&C[4]) with HM.

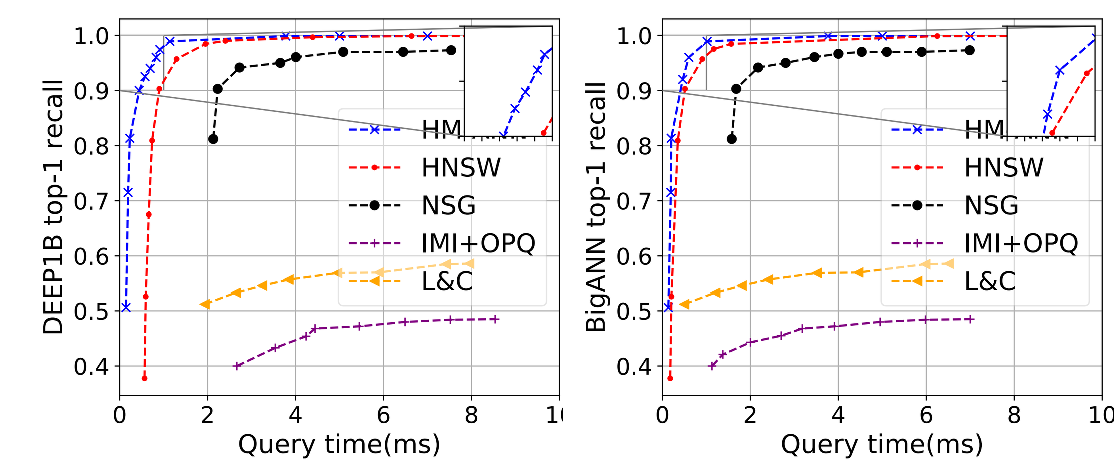
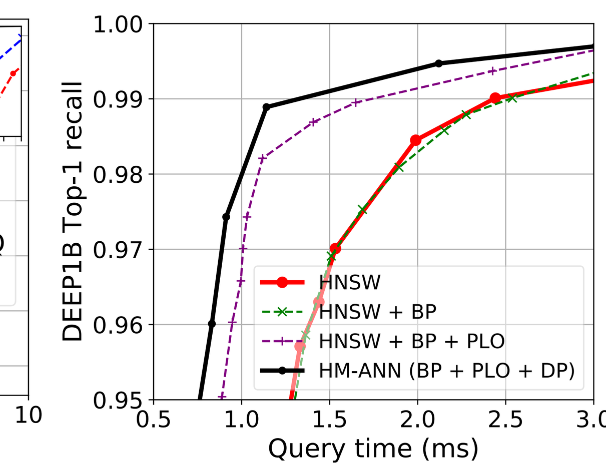


Fig. Query time vs. recall curve for top-1 recall for (a) Deep1B, and (b) BigANN



Techniques in HM-ANN

✓ HM-ANN achieves **top-1 recall** of > 95% within **1 ms**

✓ HM-ANN is **2X faster** than HM-unawared **graph-based** methods to reach the same accuracy.

✓ HM-ANN outperforms state-of-the-art **quantization-based** methods in recall-vs-latency by a large margin, obtaining **46% higher recall** under the same search latency

✓ HM-ANN achieves high search efficiency by performing Bottom-up promotion (BP), Parallel search in L_0 (PL0) and Data Prefetching (DP).

References

[1] Yury A. Malkov and D. A. Yashunin. (2020) Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*,
[2] Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. (2019) Fast Approximate Nearest Neighbor Search with the Navigating Spreading-out Graph. In VLDB'19,.
[3] Matthijs Douze, Hervé Jégou, and Florent Perronnin. (2016) Polysemous codes. In ECCV. 2016
[4] Matthijs Douze, Alexandre Sablayrolles, and Hervé Jégou.(2018) Link and Code: Fast Indexing With Graphs and Compact Regression Codes. In CVPR, 2018.
SuhasJayaramSubramanya,FnuDevvrit,HarshaVardhanSimhadri,RavishankarKrishnawamy, and Rohan Kadekodi. (2019) Rand-nsg: Fast accurate billion-point nearest neighbor search on a single node. In *Neurips 2019*