Vortex: Efficient Decentralized Vector Overlay for Similarity Search and Delivery

Shengze Wang, Yi Liu, and Chen Qian @ University of California, Santa Cruz

Contacts

OVERVIEW

- **Trends:** Sovereign/hybrid RAG (keep data in-region; multi-cloud, multi-org overlays), Collaborative/edge LLMs (planet-scale, self-organizing vector search), Serverless & disaggregated DBs (show the cost/latency pressure); Industry pull: emphasize hybrid/multi-cloud, edge, and data sovereignty for AI infrastructure
- **The problem.** Centralized or cluster-bound control planes strain cost/latency and create single point of compromise; egress costs. Latest RAG & edge/agent settings calls for self-organizing decentralized overlay. Cloud sovereignty & hybrid are rising requirements.

Can we provide ANN search with accuracy and latency comparable to the best centralized systems, while being fully decentralized, self-organizing, and planet-scale? Our answer is **Vortex**:

- **First fully decentralized ANN overlay** for AI service, delivering high-recall, low-latency search across peers without centralized control plane
- **Decentralized Learned Hashing (DLH).** We introduce a collaborative, similarity-preserving hashing scheme that maps vectors to hash space
- **Distributed HNSW (DHNSW).** Co-designed DLH→DHT→DHNSW pipeline, probing minimum peers for fast and accurate ANN delivery
- **Centralized performance, Balanced at scale, Churn-resilient, Self-organizing, Privacy-preserving, Practical deployability**

Vortex DESIGN

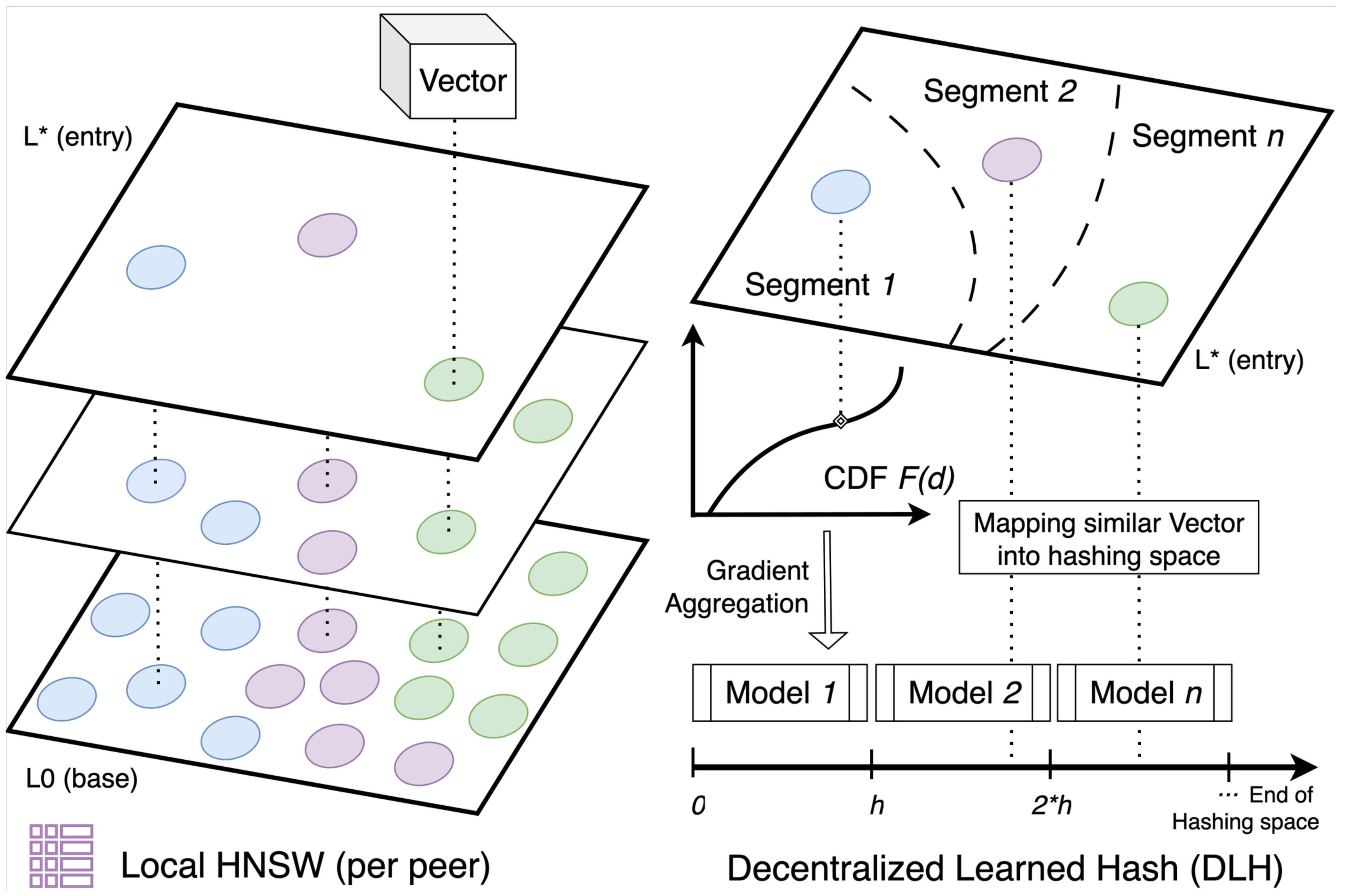

Fig. 2. Decentralized Learned Hashing (DLH): from Vector to Hash.

Decentralized Learned Hash Function (DLH) construction:

a. Learn K centroids $\{c_j\}_{j=1}^{K}$ from peers' HNSW top-layer (NSW skeleton).

b. Fit piecewise CDF $\hat{F}_j(d)$ of distances $d = \| x - c_j \|$ per centroid c_j.

c. Partition the key space into K segments with offsets and widths (o_j, w_j)

The hash is computed as : $k(x) = h\left(o_j + w_j \cdot \hat{F}_j(\| x - c_j \|)\right)$.

Distributed HNSW index (DHNSW): Vortex maintains Hierarchical Navigable Small World (HNSW) indexes over partitions for high-recall, low-latency search on each peer; Codesigned with the DLH mapping, only the peer(s) holding the query vector's likely nearest neighbors are probed.

DHT-based Routing: Vortex employs a Chord-style overlay using Distributed Hash Table (DHT) that assigns portions of the hashing space to peers for scalable, fault-tolerant routing and churn resilience; Vortex also incorporates model updates, recovery and routing optimizations from our recent work LEAD (ICNP 2025, arXiv:2508.14239, [1]).

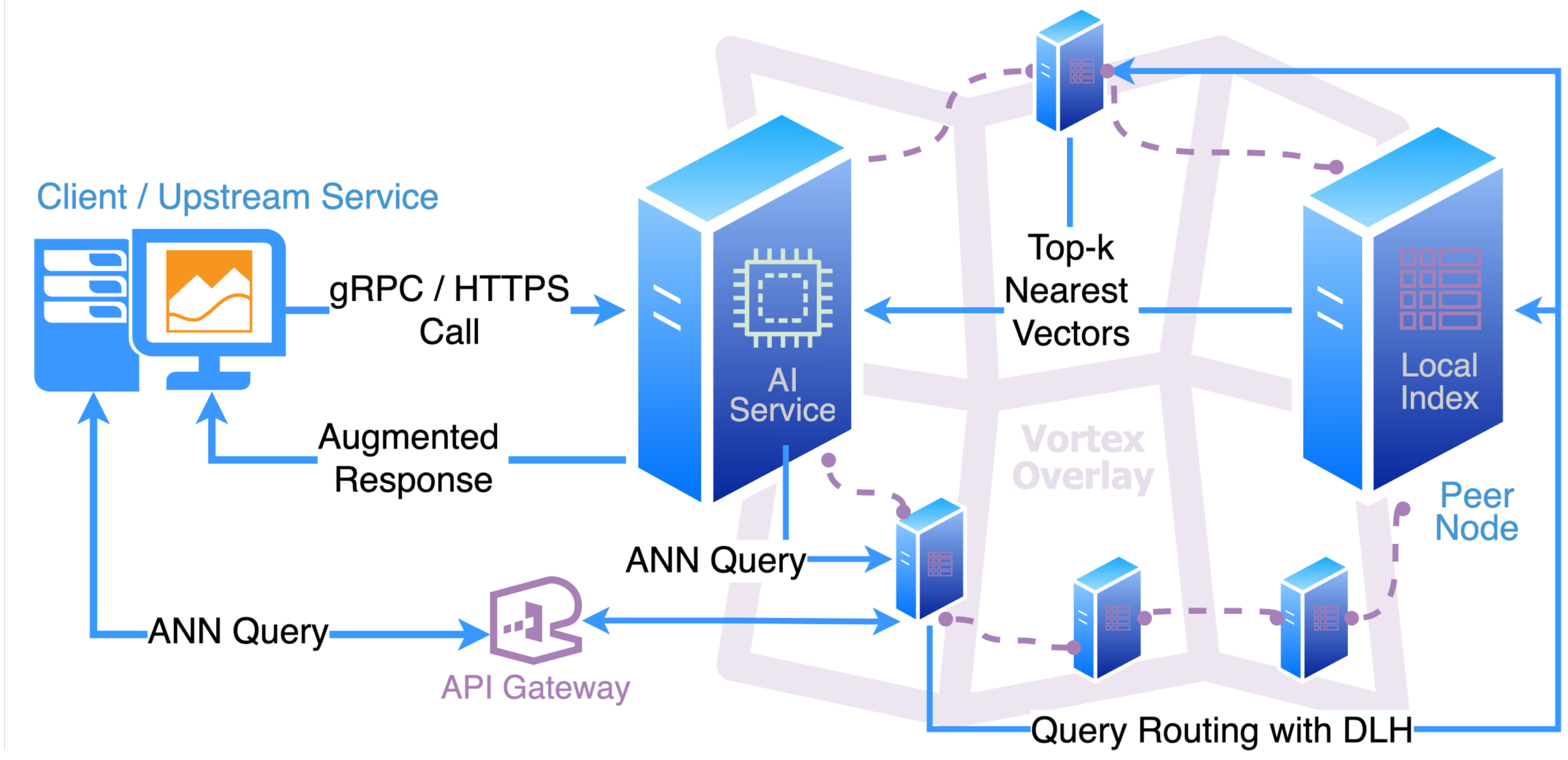

Fig. 1. Vortex system architecture and ANN query path.

PRELIMINARY RESULTS

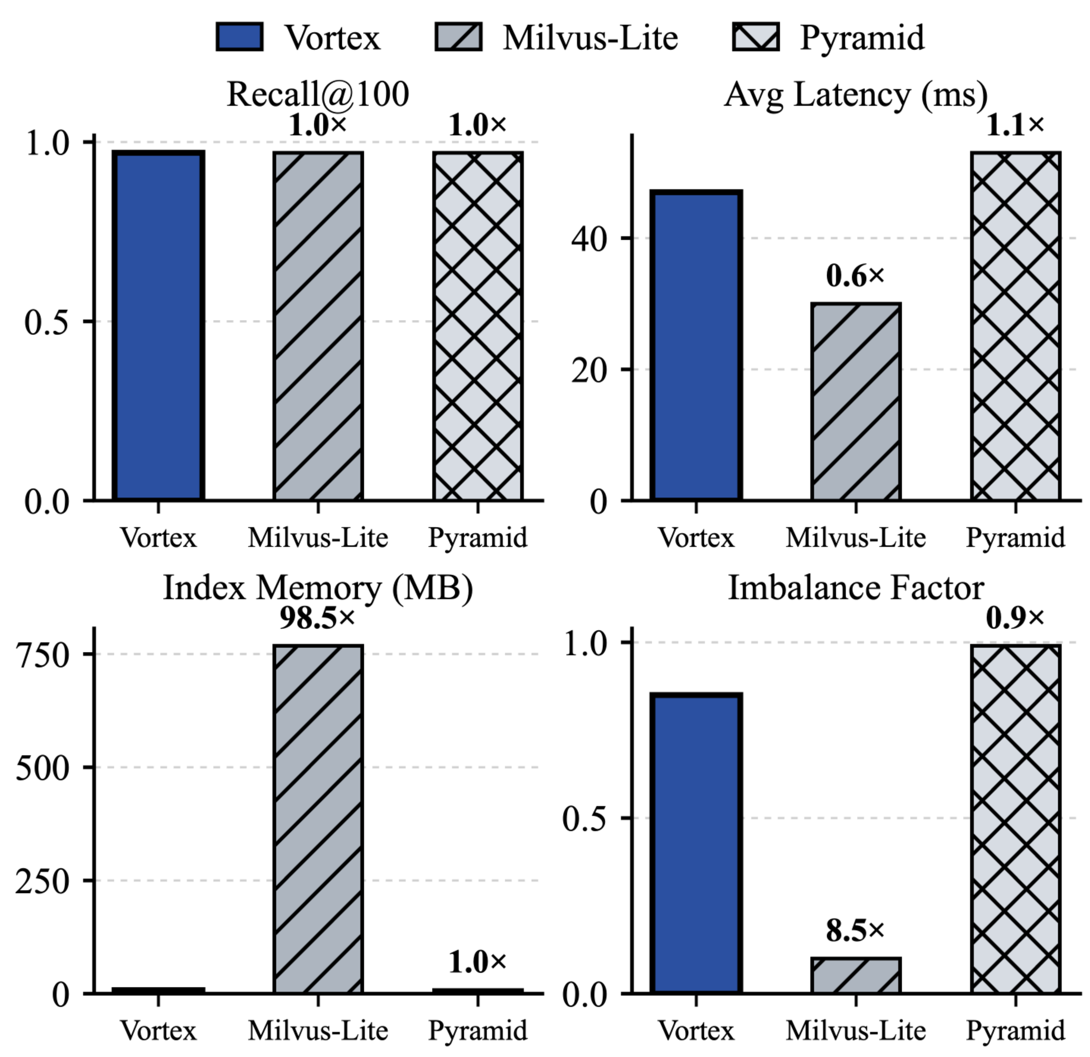

Fig. 3. Benchmark of Vortex on SIFT1M@100.

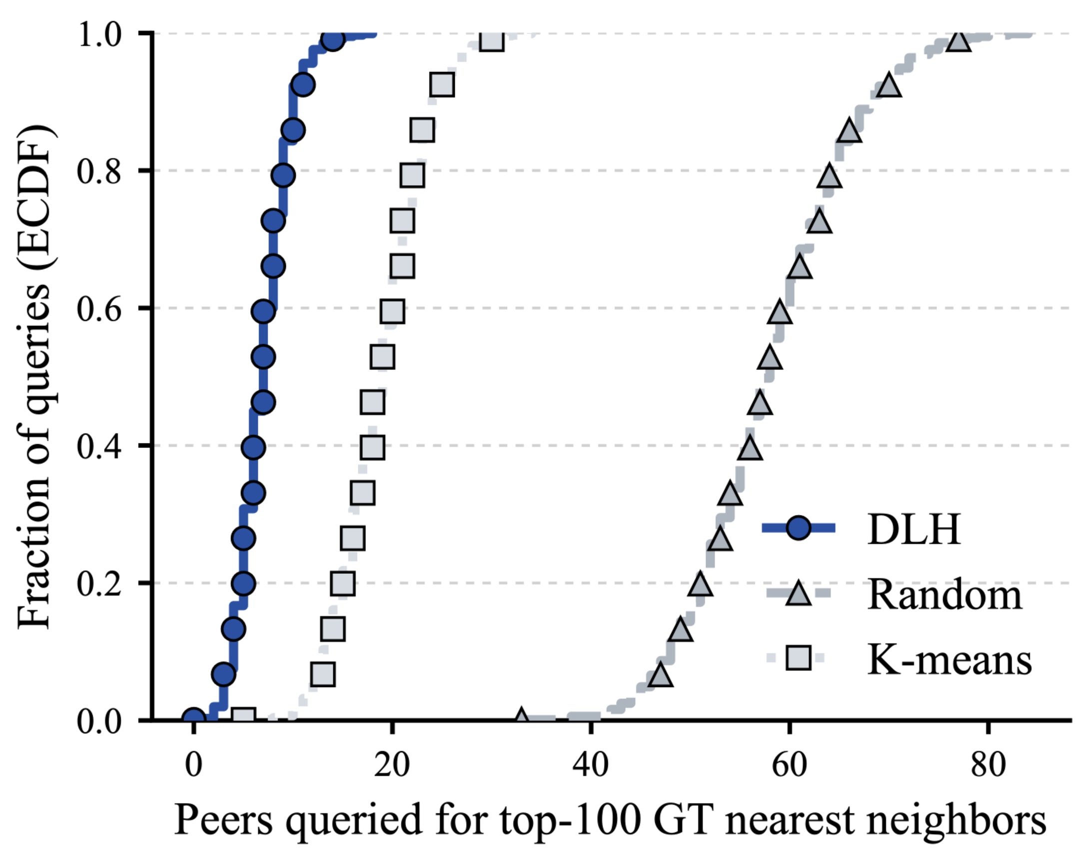

Fig. 4. Peers probed to cover Ground-Truth@100.

Baselines:

- *Milvus-Lite*, lightweight variant of SotA single-cluster Milvus system [3]
- *Pyramid* [4], coordinator based framework partitions vectors via k-means, builds per-shard HNSWs, and probes the top-t shards at query

Observations :

- *Vortex* matches the Recall@100 of SotA centralized baselines, delivers competitive average latency
- *Vortex* probed ≤10 peers out of 100 peers for 95% of queries and cut per-peer index memory by ~100**×**.
- *Vortex* maintains a more balanced load than other baselines.
- *Vortex* provides the benefits of autonomy, fault tolerance, and data-sovereignty.

CONCLUSION

- **Vortex reframes the design point for VectorDB/Overlays.** Vortex demonstrates centralized-class accuracy/latency is achievable without centralized control or storage, opening a new path for AI infra where sovereignty, resilience, and cost are first-class alongside performance.
- **Decentralized Learned Hash-driven key space and overlay-native model updates** provide a pattern for learned distributed systems.
- Vortex is part of our broader effort to **democratize LLM deployment & open serving**; alongside it, we introduced GenTorrent [2], and Vortex is designed to work natively with such collaborative LLM systems.

References

[1] Wang, S., Liu, Y., Zhang, X., Hu, L. and Qian, C., A Distributed Learned Hash Table. arXiv preprint arXiv:2508.14239. 2025.
[2] Fang, F., Hua, Y., Wang, S., et al., GenTorrent: Scaling Large Language Model Serving with An Overley Network. arXiv preprint arXiv:2504.20101. 2025.
[3] Wang, Jianguo, et al. "Milvus: A purpose-built vector data management system." Proceedings of the 2021 international conference on management of data. 2021.
[4] Deng, Shiyuan, et al. "Pyramid: A general framework for distributed similarity search on large-scale datasets." 2019 IEEE International Conference on Big Data.
[5] Wang, Jingdong, et al. "A survey on learning to hash." IEEE transactions on pattern analysis and machine intelligence 40.4 (2017): 769-790.
[6] Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in neural information processing systems 33 (2020): 9459-9474.
[7] Malkov, Yu A , et al. "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs." IEEE transactions on pattern analysis and machine intelligence 42.4 (2018): 824-836.

Baskin Engineering UC SANTA CRUZ