SDC | StorageAI™

A SNIA Event

April 29, 2026 • Denver, Colorado

DiskANN Indexing Offloading

Alessandro Goncalves
Harsha Simhadri

Agenda

- Vector Database
- Embeddings
- ANN Similarity Search
- Index Types
- DiskANN
- Key Results
- Next Steps

Vector Database

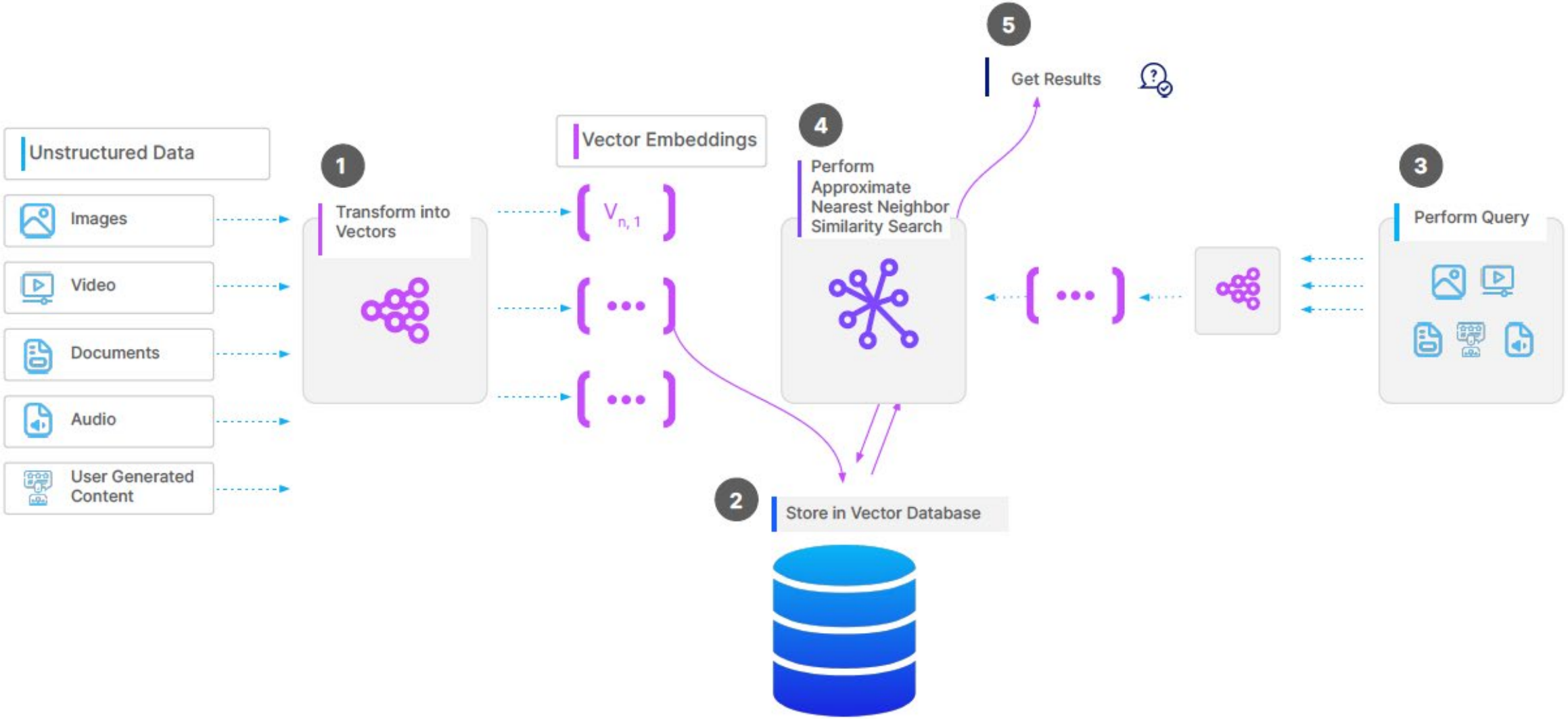
Purpose-built databases meant to conduct approximate nearest neighbor search.

Search across a large dataset of high-dimensional vectors.

Vectors meant to represent semantics of unstructured data.

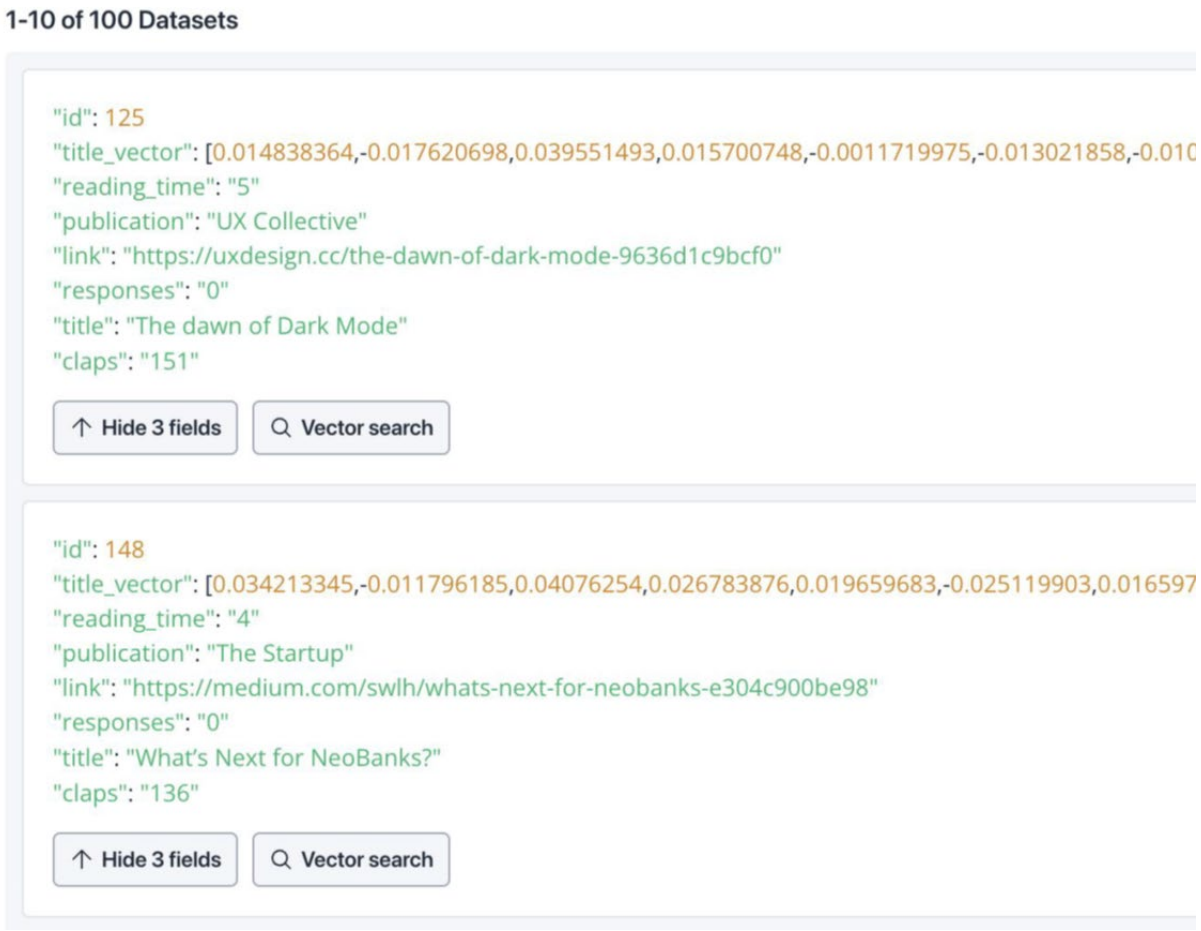
Similarity search requires a data structure known as vector index.

ANN Similarity Search

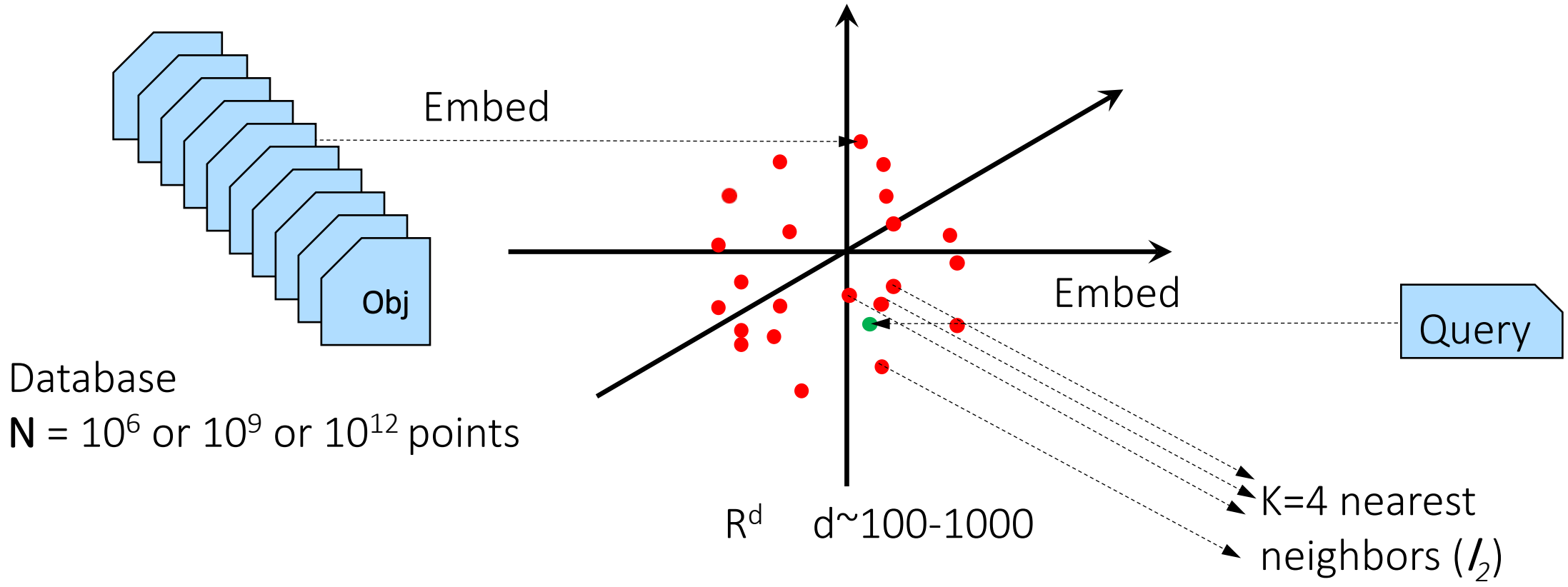


Embeddings

- Numerical representation of data.
- Mapping a high dimensional input into vector space.
- Provide Semantic Similarity.



Semantic retrieval with embeddings + ANNS



Exact retrieval might need exhaustive scan, settle for approximate retrieval.
Measure **Recall@k**: fraction of output candidates that are true top-k nearest neighbors

Index Algorithms

- Flat
- Tree-Based
- Graph-Based
- Clustering
- Quantization
- Hybrid

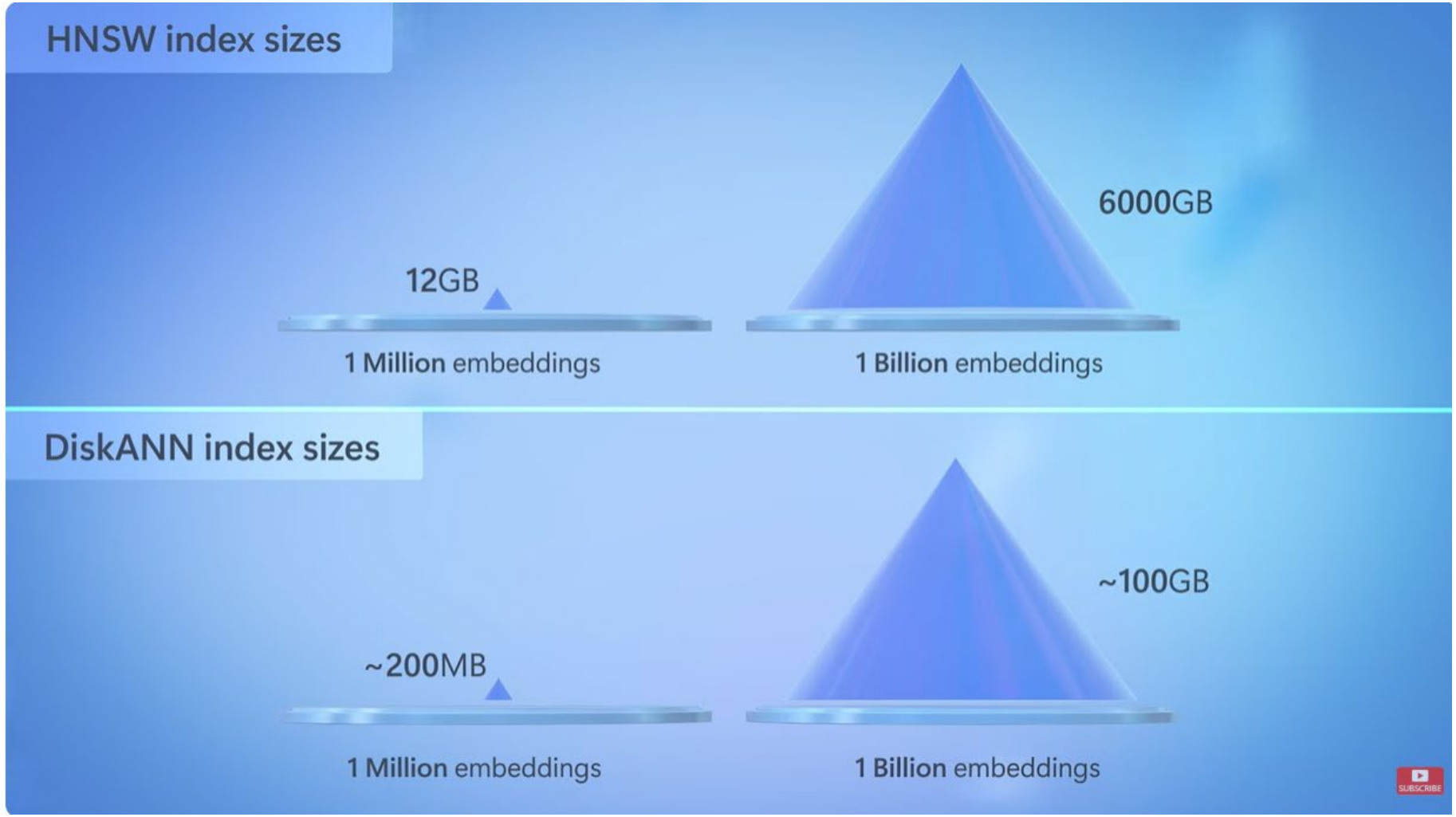
Index Footprint

- Searches happen via the Index.
- Several Indexing types available.
- Sizing based on:
  - Number of vectors
  - Dimension
  - Bytes per component
    - Float32 (4 bytes) , float16 (2 bytes), int8 (1byte)

Index Memory Sizing

- Graph-Based (HNSW)
  A. Graph Structure - #Vectors * 4 bytes (INT32)
  B. Raw Vector Embeddings - #Vectors * #Dimensions * 4 bytes
  C. A+B , (not considering quantization)
- DiskANN
  - Vamana graph stored on disk
  - Compressed Vectors in Memory

Index Memory Usage (HNSW x DiskANN)



1 Billion Vectors / 768 dimension

- Using Milvus sizing tool
  - https://milvus.io/tools/sizing
- Operational Cost to support 1B vectors / 768 dim

	Cores	Memory	Storage	Local Disk	Cloud Cost
HNSW	1000	3.86 TB	33.87 TB	0	$23K
DiskANN	315	1.17 TB	23.21 TB	3 TB	$9K

Zilliz Cost Estimator Sep.2025

Vector Search at Microsoft

@Microsoft	Cloud			Edge
	Web Search, Ads & Recommendations	**Enterprise Email Search**	**Enterprise Doc search**	**Windows Copilot Runtime**
Index Size	100s of billions of pages	Millions of indices 10^{14}+ sentences	Thousands of indices Trillions of paragraphs	~10^6 files, images, text
Update Rate	~ billion /day real-time updates	new email, clean up deleted emails	~1% change/day	As new content as created
Search latency and throughput	~10ms latency 10^4-10^5 queries/sec	<100ms, <100 queries/day	<100ms, 100 queries/sec	<100ms, <100 queries/day

Very large range of sizes, ingest and query throughput.

DiskANN – Hybrid Graph Based Index

- DISKANN builds k-NN graphs (like HNSW) and it is a graph-based index.
  - Memory + Disk Hierarchy
  - Memory – Medoid Graph
  - Disk – Resident Flat Graph
- Optimized Disk Layout
  - Offline reordering
  - Graph connections stored in a sequential layout.
- https://github.com/microsoft/DiskANN/

DiskANN: High recall, low latency via hybrid DRAM+SSD index

[Subramanya, Devvrit, Kadekodi, Krishnaswamy, Simhadri'19]

DRAM

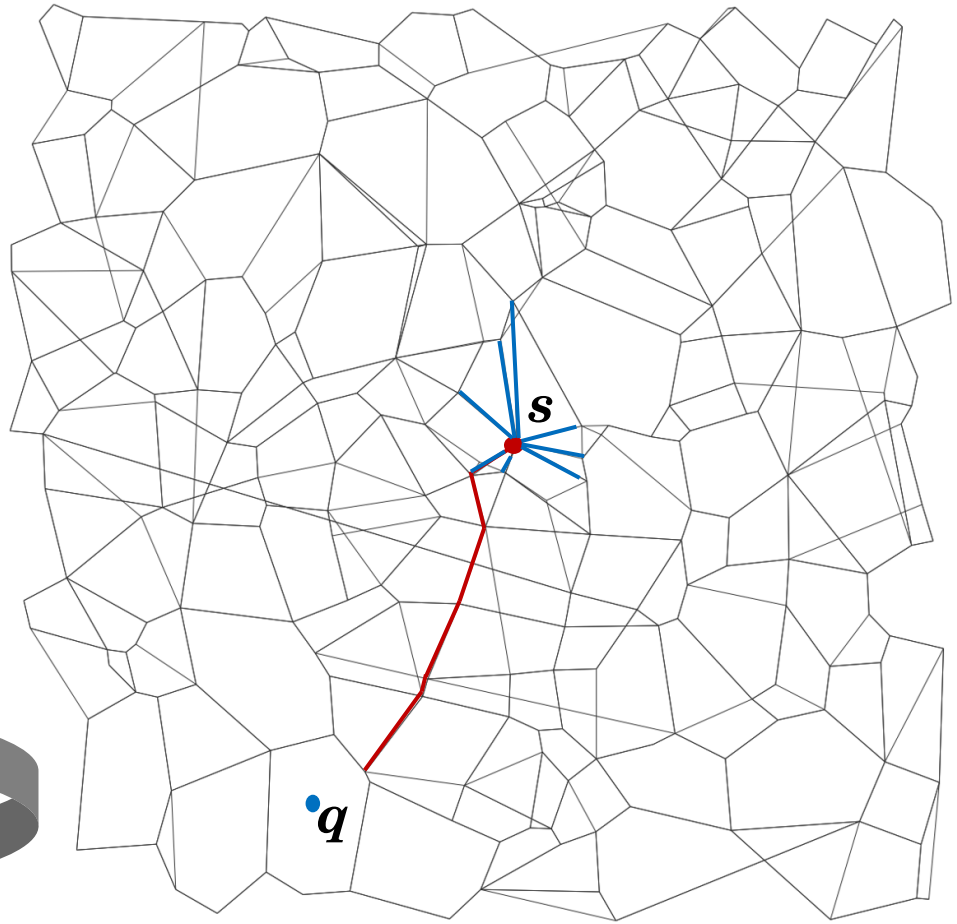
Compressed vectors (~32B)

SSD

1B Vectors (100-1000d) +
Graph(~100 degree) ~ 500GB-1TB
Low Diameter (<10 hops)

GreedySearch(*q*)

- Let *p := s* (start node)
- Fetch neighbors of p from SSD
- Use **compressed representation of points** to find neighbor *p* closest to *q*

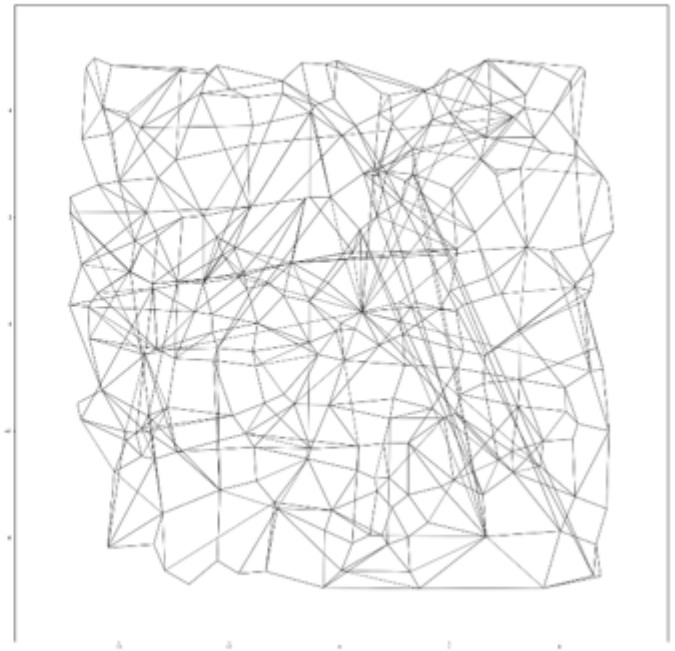


DiskANN Graph

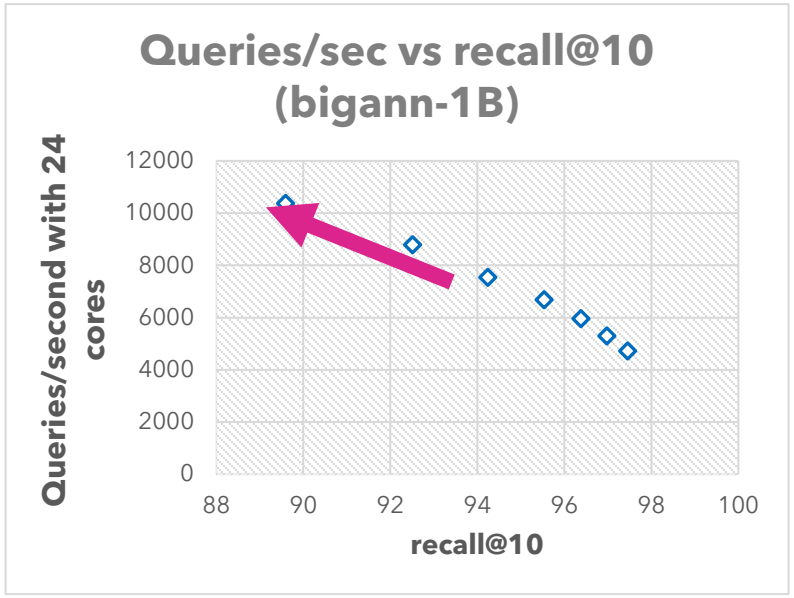
- Mix of nearest neighbors
- Diverse long-range edges for fast navigation

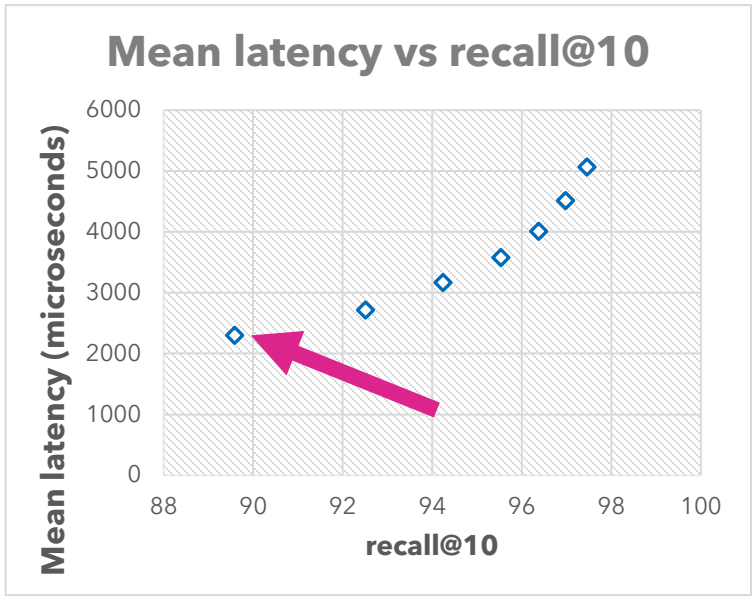
α=1

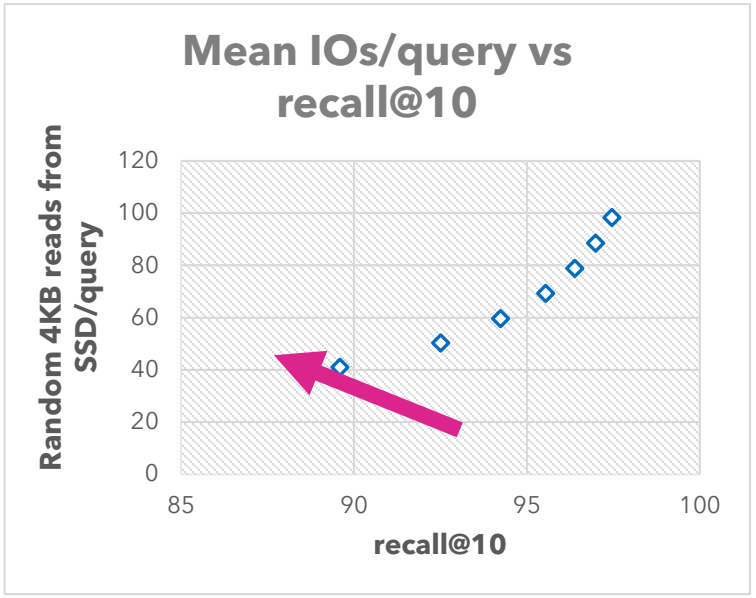
α=1.2



Recall, latency, QPS and IO/s for 100-degree graph







BIGANN dataset: 1Billion points in 128 dimensions

Memory footprint = 32GB + 250K adjacency lists cached in memory ~ 33GB

In comparison, in-memory graph indices (e.g. HNSW) would need 500GB+ DRAM

The DiskANN project: Algorithmic Innovations

Problem 1: Serving O(100B) indices with <10ms latency and high throughput requires in-mem indices requires 10,000s of machines with O(100GB) DRAM

→ DiskANN [NeurIPS'19]:

Index ~1B vec/machine using SSDs; 5-10x higher density than in-mem; <10ms query latency, 10000+ QPS.

Problem 2: Hard to update, deletes with stable recall is hard. Rebuilt from scratch periodically. 10,000s of machines to periodically rebuild indices every 6/12/24 hours.

→ Fresh-DiskANN[arXiv:2105.09613]
IP-DiskANN[arXiv:2502.13826]

DiskANN + Real-time freshness + 1000s updates/sec/node

Problem 3: Not designed for filtered/hybrid queries. Low recall and high query complexity.

```
e.g., ORDER by L2 distance
      AND (date > Aug 2023)
      AND (brand = x OR y OR z)
```

→ Filtered-DiskANN [WWW'23]:

Order(s) of magnitude higher QPS and lower query latency;
High recall for rare predicates.

DiskANN & CosmoDB Key Results

Performance

- **<20ms query latency on 10M vectors.**
- **< 2X increase in latency/cost as index scales from 100D->768D, 100K->10M vectors.**

Stable Recall and Latency

- **Stable Recall over long stream of updates even with distribution drift.**
- **No ingestion or query latency spikes due to rebuilds or segment merges.**

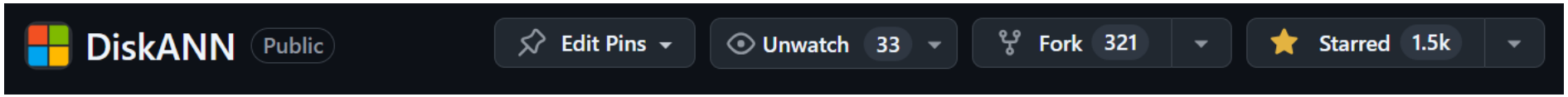
Filtering and Multi-tenancy

- **Filtered queries with upto 8X better P99 cost and latency with new algorithms.**
- **Native Multi-Tenancy: Fast Search and High Recall on a long tail of tenants.**

Cost and Scale

- **Scale out to 1 Billion Vectors with automatic partitioning.**
- **Lower query costs. Upto 43X and 12X than Pinecone and Zilliz respectively at 10M scale**

OSS Release (~2023) and Industry Impact



Microsoft

- Bing web search, real-time index
- Advertisements
- Microsoft 365
- Windows

Relevance lifts, new scenarios enabled, and hardware saved

Adoption in other databases

- Postgres-> TimescaleDB/Pgvectorscale
- Cassandra ->IBM/Datastax
- SQLite -> Turso
- Milvus and other vector DBs.

Hardware/Accelerator

- Kioxia AiSAQ for storage-only solution
- Intel OptaNNE for Optane pmem
- NVIDIA cuVS library

DiskANNv3
In-process stateless library for any DB

One branch for Research and Production;
Amplifies reach of new ideas;
As fast as monolithic in-mem & on-disk libs

User/ Indexer/ Query Engine

Update

Query

DiskANN indexing and search algorithms (stateless)

Update

Read

Quantized VectorProvider

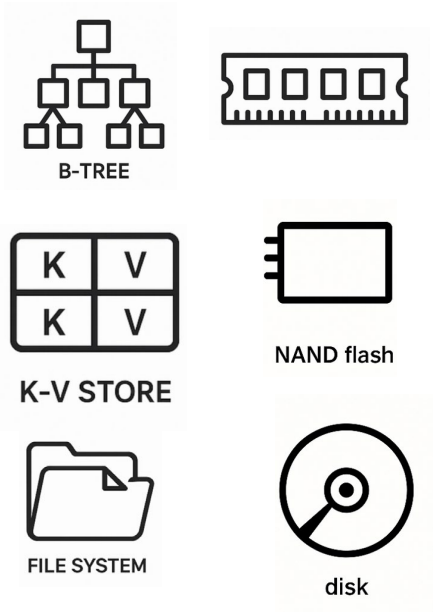
Vector Provider

Edge List Provider

Label/Metadata Provider

Encode

Read



Operate at multiple storage tiers

DiskANN library provides advanced index & query logic and quantizers

Delegate storage of index terms to DB and access/mutate via Provider abstractions

Inherit elasticity, availability and durability of host DB

New Algorithmic Challenges

- One Graph – incrementally updated, no merges, no rebuilds
  - [Xu, Manohar, Bernstein, Chandramouli, Wen, Simhadri'25]
- Minibatch updates for parallel, deterministic single-writer update
  - [Manohar, Shen, Blelloch, Dhulipala, Gu, Simhadri, Sun'24]
- Ingest in quantized space for low cost and high performance
- Faster algorithms for vectors + arbitrary predicates that work closely with the database's query planner


SDC | StorageAI™
A SNIA Event
Thank You