Lowering TCO of Vector Search in AI With Flash-Driven Efficiency

A SCALABLE, COST-EFFICIENT ALTERNATIVE TO CURRENT ANN SEARCH ARCHITECTURES

[Confidential]

08.05.2025

VISHWAS SAXENA, SENIOR TECHNOLOGIST

AABHA MISHRA, SENIOR ENGINEER

CONTENTS

(→)

ANN Search & Relevance to AI/ML Applications

Approximate Nearest Neighbor (ANN) search is a fast way to find items that are most similar to a given query, without checking every item one by one.

Large-Scale Applications of ANNS in AI/ML

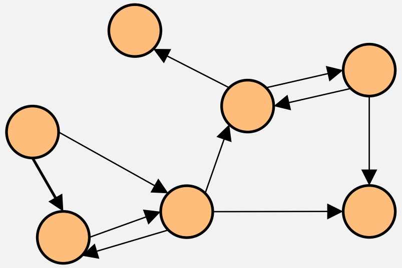
Algorithms for ANN Search

HOW IT WORKS	**ALGORITHM**	LIMITATIONS
Compute distances to all points	Naïve Linear Search	Brute-force; slow — O(nd)
Recursively split space into smaller, searchable regions	Space Partitioning	Works well in low dimensions; fails in high-dim (curse of dimensionality)
Hash similar points to the same bucket; query via hash collisions	Locality Sensitive Hashing	Fast to build, but slower to query than newer methods

Graph Based Indexing

Vectors are **nodes** connected to nearby vectors through **edges** based on vector similarity.

- High accuracy across a wide range of datasets
- Low-latency query performance



- Scalability to handle large collections
- Ability to operate across diverse data types

SOTA Graph Implementations – NVIDIA’s CAGRA

FEATURES

Uses a pruned KNN graph, optimized for GPU memory and traversal

Fully GPU-resident index for in-memory access

WORKFLOW

Thousands of CUDA threads explore and evaluate candidates in parallel

Batch-query optimized for high-throughput workloads

PROVISIONS

Enables fast, large-scale ANN search with high parallelism

Suitable for real-time inference and LLM retrieval scenarios

LIMITATIONS

Requires the entire index to fit in GPU memory. Performance drops sharply otherwise, making it expensive.

SOTA Graph Implementations – Microsoft’s DiskANN

FEATURES

Built on NSG: a pruned KNN graph enabling sparse, efficient traversal

Stores full-precision vectors on disk, compressed data in RAM

WORKFLOW

Disk-based search with I/O scheduling and multi-threaded traversal

Employs some limited CPU parallelism

PROVISIONS

Enables billion-scale search using commodity CPUs and SSDs

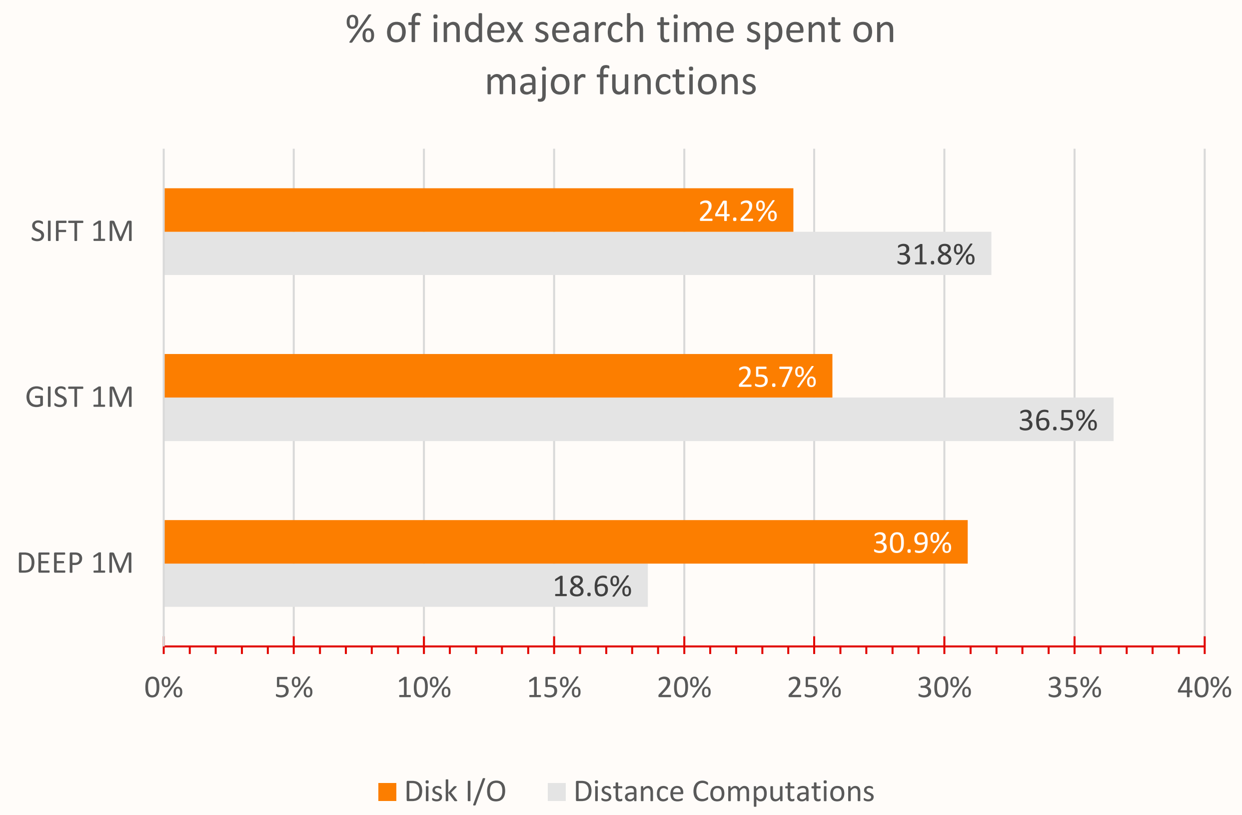
Does not require high in-memory capacity — cost-efficient scalability

LIMITATIONS

Higher latency than in-memory approaches due to disk access

I/O-bound: full vector data must be retrieved from disk during search

DiskANN Profiling & Conclusions



OBSERVATIONS

30% spent on distance comps.

27% spent on disk I/Os

CONCLUSIONS

To optimize latency & throughput:

- Move search process closer to data
- Accelerate distance computations

Our Contribution @ SanDisk

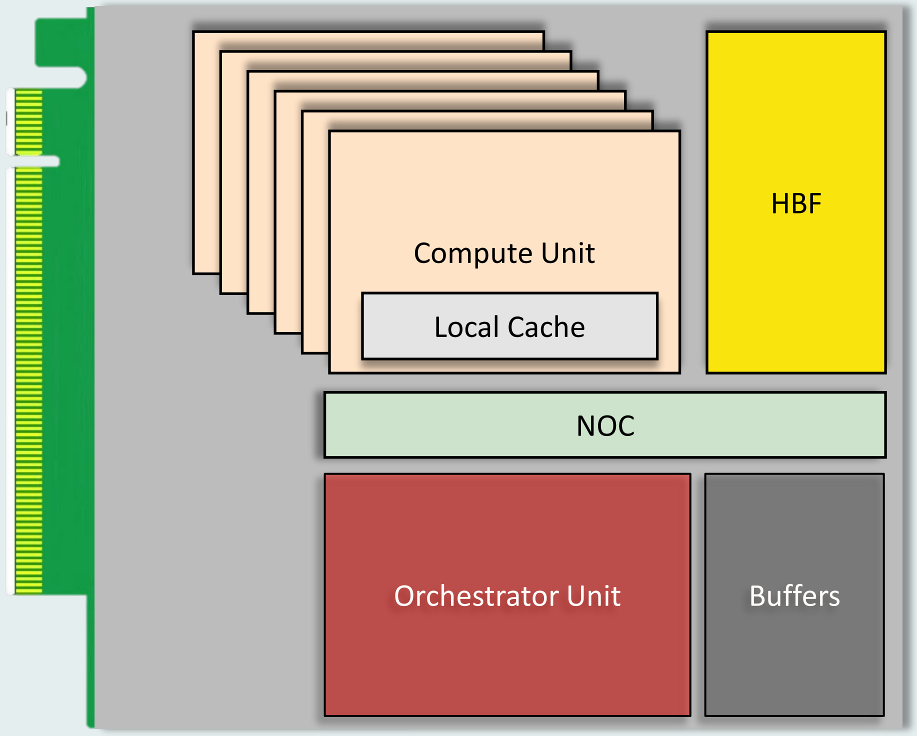
A central CPU orchestrates and coordinates multiple specialized compute units.

Compute units execute core vector similarity operations using efficient processing arrays.

Hardware-accelerated structures handle rapid top-k selection of results locally within compute units.

Each compute unit includes local memory caches to optimize data access and throughput.

The central CPU manages data transfer between a high-bandwidth persistent memory, on-chip buffer, and compute units.



SanDisk HBF Press Release

Benefits Over Market Offerings

Over CPU Solutions

Parallelism

Multiple compute units operate concurrently, vastly increasing throughput compared to sequential CPU execution.

Specialized Hardware

Flow of data through the compute units is completely in hardware path, increasing speed per similarity computation beyond CPUs.

Lower Power Consumption

Custom hardware tailored for specific tasks achieves better performance-per-watt than general CPUs running the same workload.

Reduced Data Movement

Local caches and direct memory access minimize latency and bandwidth bottlenecks inherent in CPU memory hierarchies.

Over GPU Solutions

Lowered TCO Per Search

By storing the graph index in high-bandwidth flash (HBF), our solution significantly lowers memory and compute cost per search without sacrificing scale.

Summary & Looking Ahead

- **ANN search** is critical for AI/ML workloads, powering applications like RAG, recommendation, and semantic search at scale.
- **Graph-based indexes** are the state-of-the-art due to their accuracy, scalability, and efficiency.
  - **NVIDIA’s CAGRA** offers high-throughput GPU search but is bound by in-memory space & costly at scale.
  - **Microsoft’s DiskANN** enables large-scale CPU-based search on commodity hardware but suffers from I/O latency.
- **Our solution** leverages High Bandwidth Flash (HBF) with custom compute to deliver:
  - Lower **TCO per search** by offloading index storage from in-memory
  - Higher **parallelism and throughput** through specialized hardware
  - Better **scalability and efficiency** than current solutions
- Looking ahead, we aim to extend support for index build algorithms as well.

THANKS!

SANDISK™