AWS re:Invent

DECEMBER 1 – 5, 2025 | LAS VEGAS, NV

Session: ANT213

Build GPU-boosted, auto-optimized billion-scale VectorDBs in hours

Dylan Tong

Product Lead, AI and vectors

Amazon OpenSearch Service

AWS

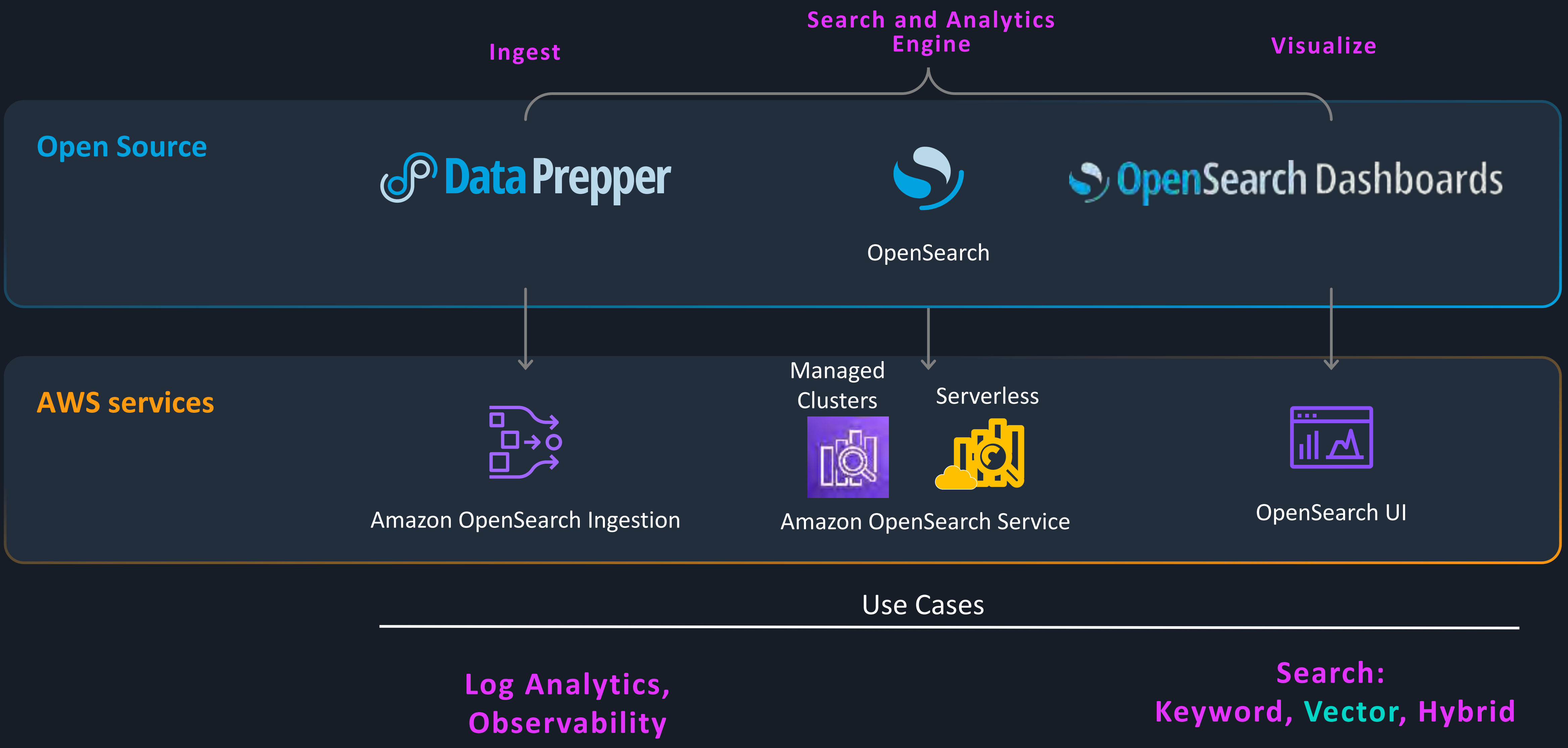
Vamshi Nakkirtha

Senior Manager, Software Dev, Vector Search

Amazon OpenSearch Service

AWS

OpenSearch: AWS and Open Source



Why Vector Search?

1. Improves search quality (relevance)
2. Versatile support for content types
3. Diverse Use Cases
4. Agentic (AI) App Enabler

Vector Search Journey

Feb. 2020

First Release

2021

OpenSearch 1.0:
(Elastic 7.10 and
Apache Lucene Fork)

FAISS

2023

Serverless

Amazon Bedrock KB

Hybrid Search

AI Native:
Vector Generation

2024

Cost Optimizations:
Automated Quantization
Tiered Vector Storage
Disk-optimized Engine

AI Native:
AI enrichments

2025

Cost Optimizations:
S3 Vectors Integration
Intel AVX-512
Graviton Neon
Storage Reduction

AI Native:
Search AI and Agentic Flows
Automatic Semantic
Enrichment
MCP Server Integration

Empower Scale:
Managed Vector Ingestion
GPU-accelerated Vector
Indexing
Vector Auto-optimize

Customers trending to billion-scale and beyond

IP Infringement Detection

8 billion attempted changes to product detail pages for signs of potential abuse (2022)

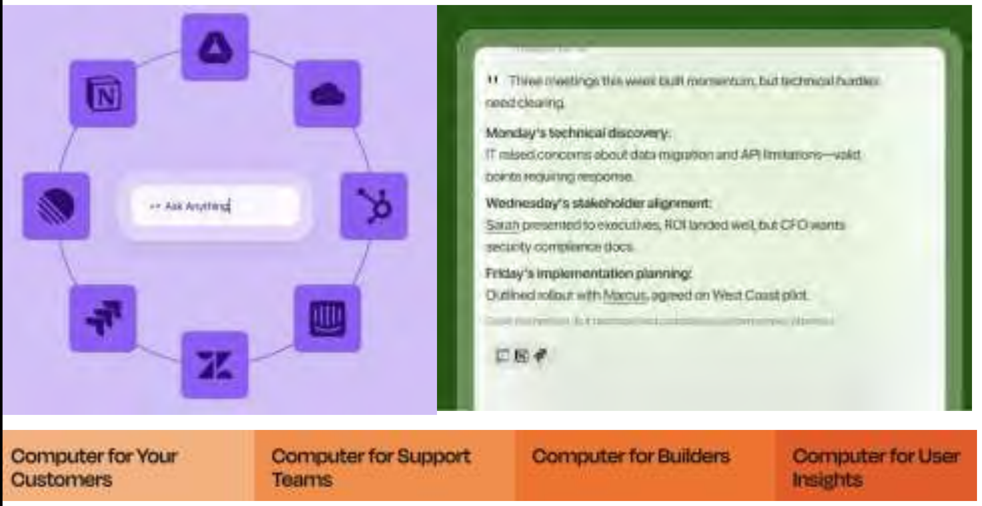
68 Billion vectors encoded from product information indexed into OpenSearch to power vector search.

99% of discovered infringements were automatically found or blocked through proactive controls

Source: https://aws.amazon.com/blogs/big-data/amazon-opensearch-services-vector-database-capabilities-explained/

Agentic Application

DevRev



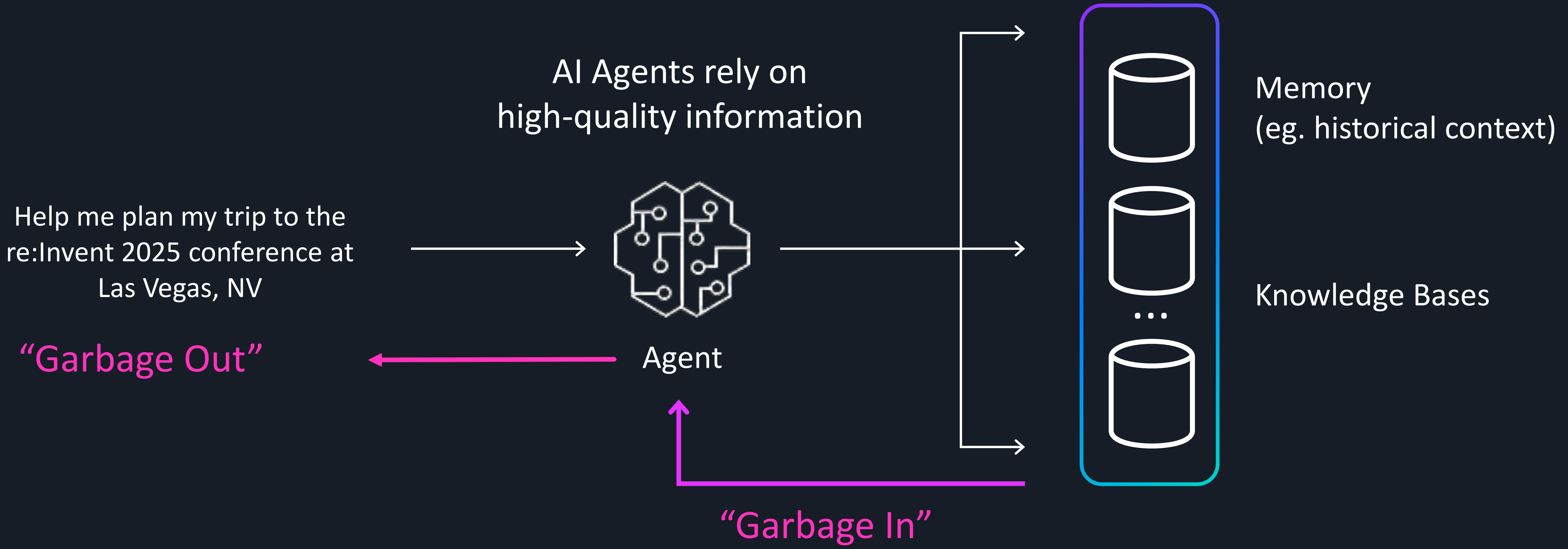
AI teammate that **powers human and AI collaboration.**

Problem:
Silo data and tools across support, product and sales teams

Solution:
Unify, automate, search and insights via AI agents.
Hundreds millions of vectors with **~1M vector updates daily.**

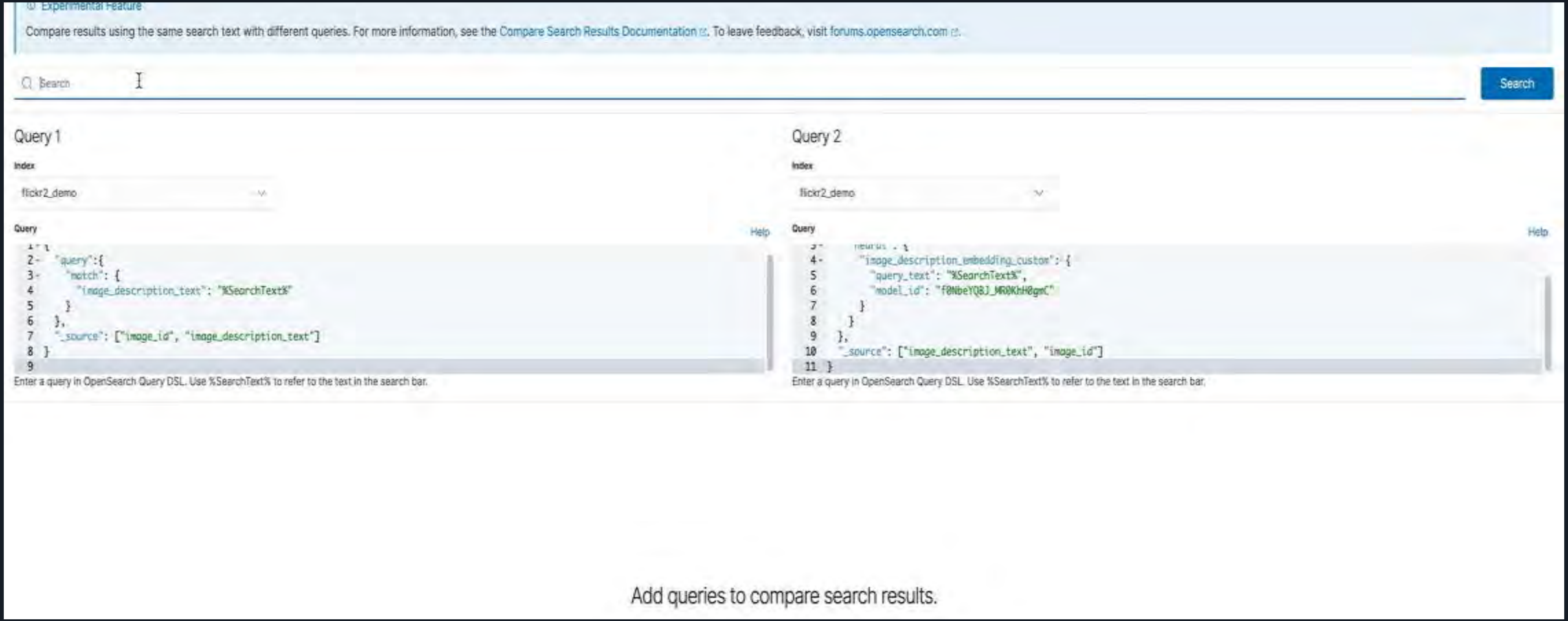
Results:
85% of tickets resolved with no human intervention,
50% cost reduction in customer support
10-hours saved per employee every week.

Agentic systems rely on high-quality search

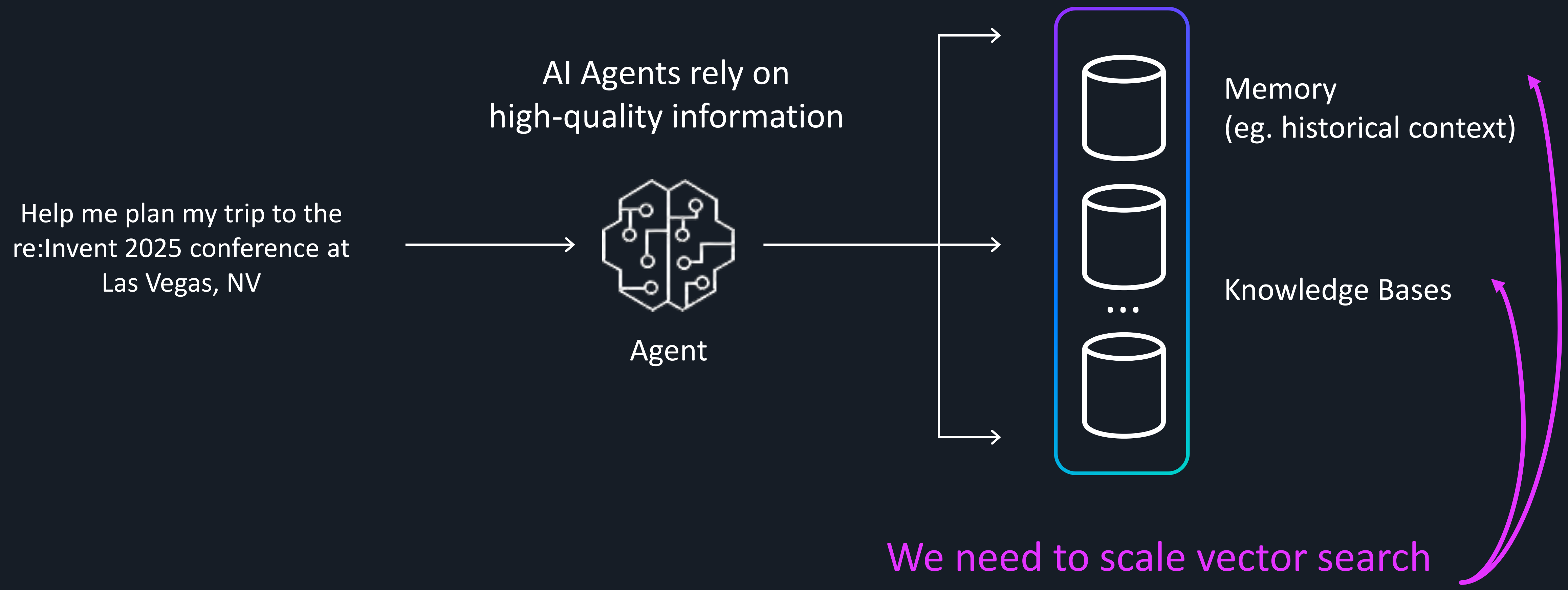


Vector search and hybrid variations are state-of-the-art.

Keyword vs Semantic (Vector)



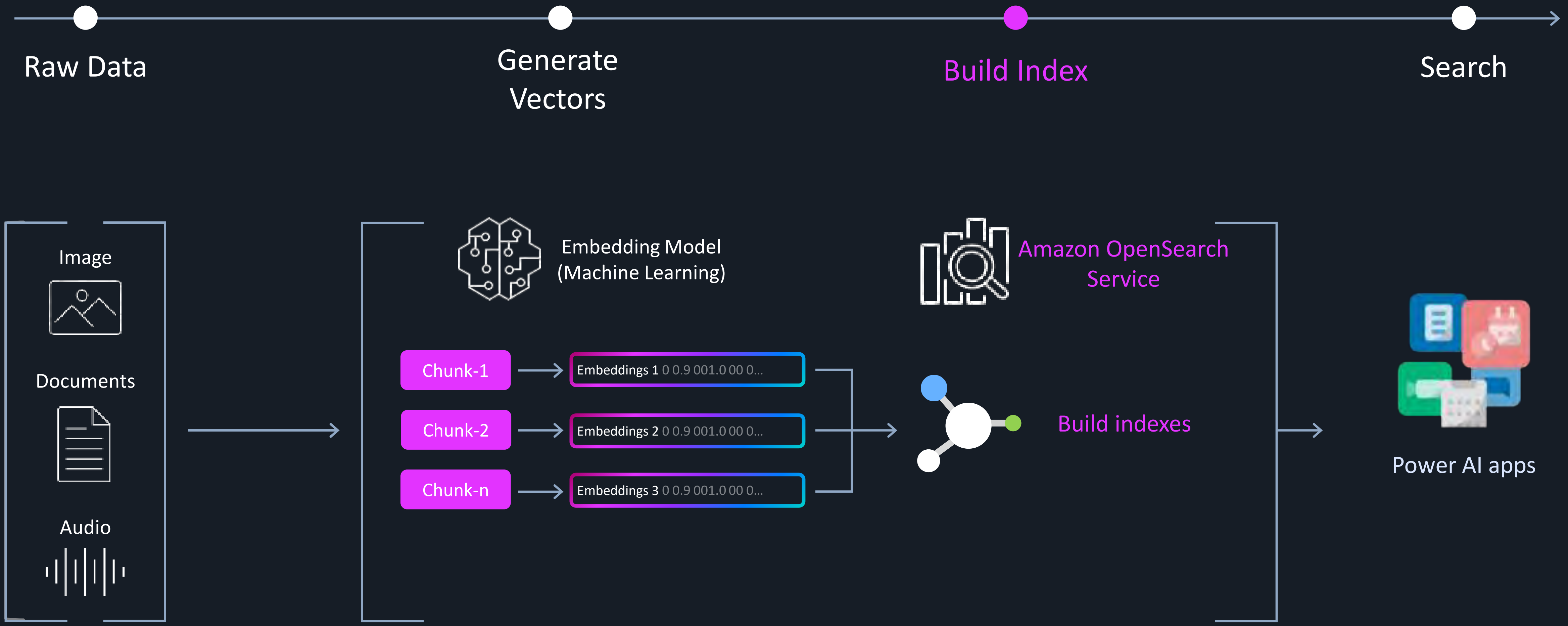
Agentic systems need vector search across vast knowledge bases…



We need to scale vector search
across Enterprise data sources

Big Vector Databases, Big Challenges

Building a Vector Database



Vectors

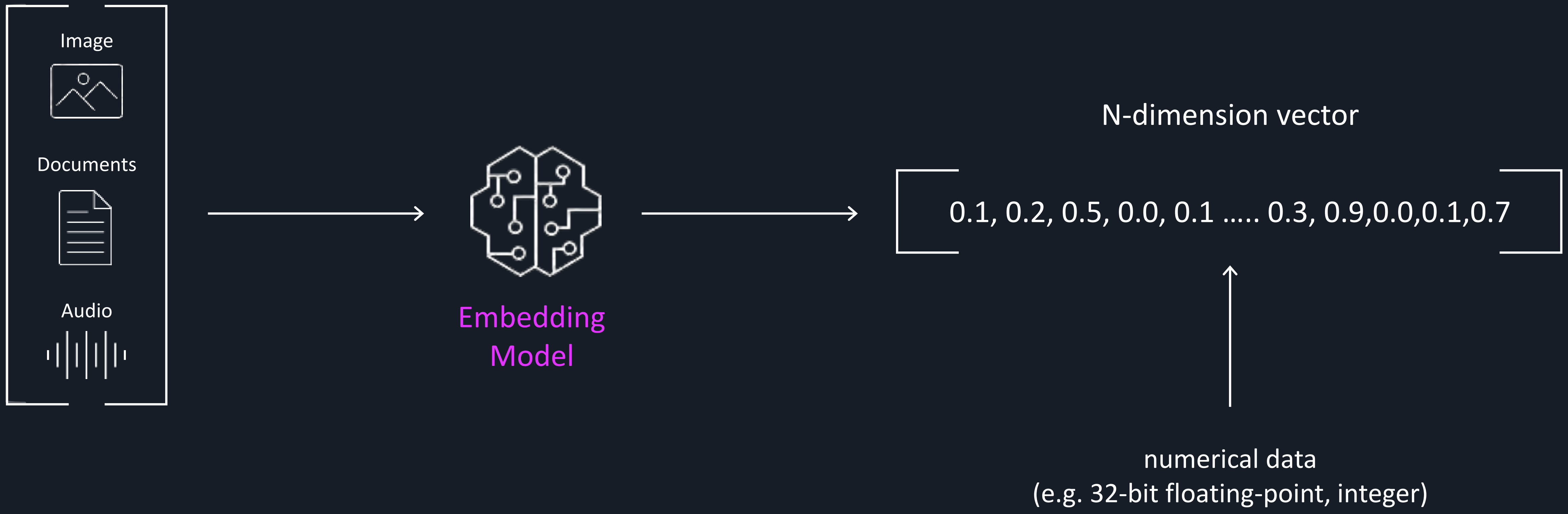
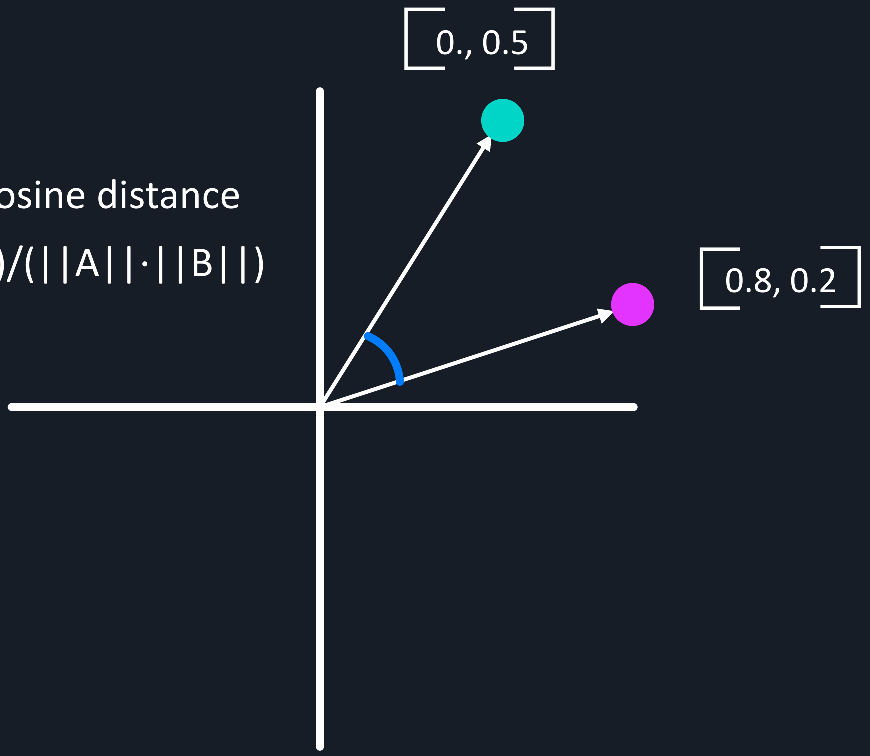


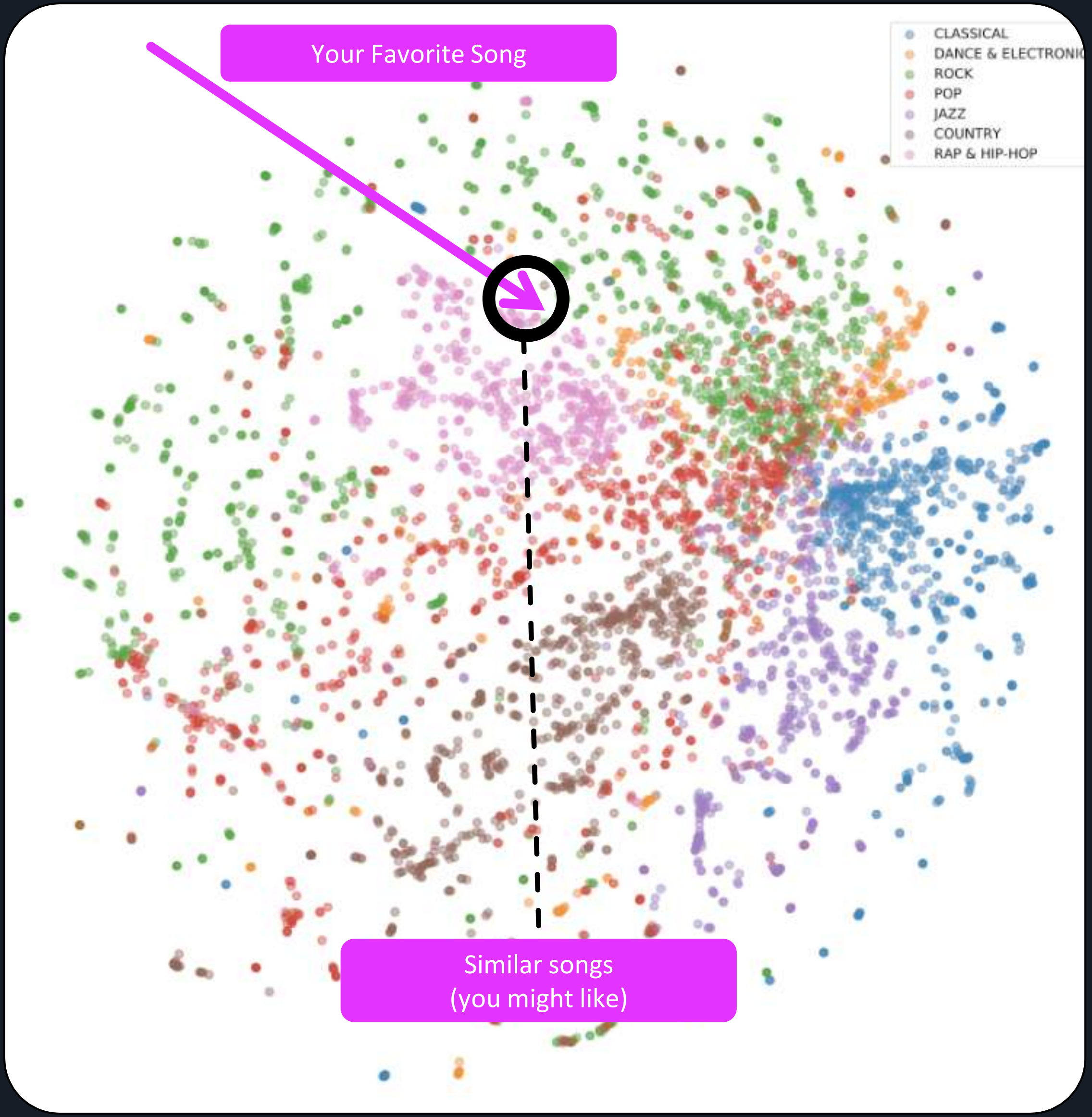
Image
Documents
Audio
Embedding
Model
N-dimension vector
0.1, 0.2, 0.5, 0.0, 0.1 ….. 0.3, 0.9,0.0,0.1,0.7
numerical data
(e.g. 32-bit floating-point, integer)

Content (Vector) Similarity

e.g. cosine distance

1−(A·B)/(||A||·||B||)





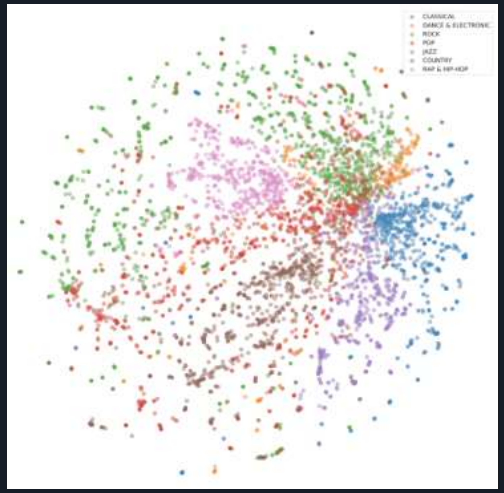
k-Nearest-Neighbors: Find the “Top K” most similar…

Exact (Brute-force) k-NN

query vector: [0.1, 0.2, 0.6 …. 0.3,0.7]
(e.g. your favorite song)

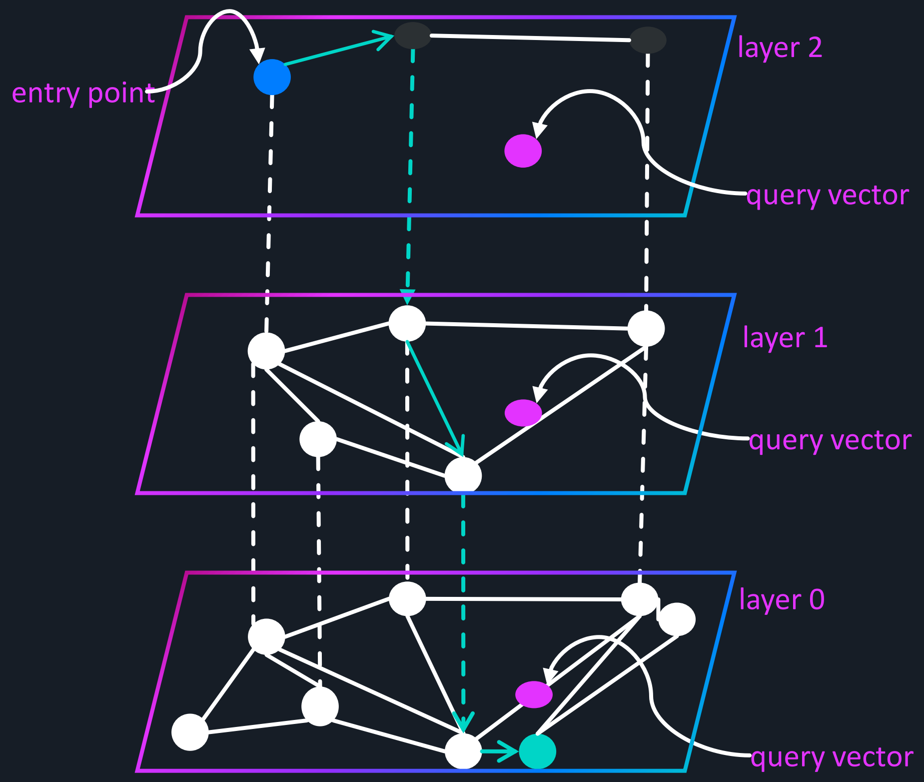
Calculate distance and rank ↕

Song vector corpus



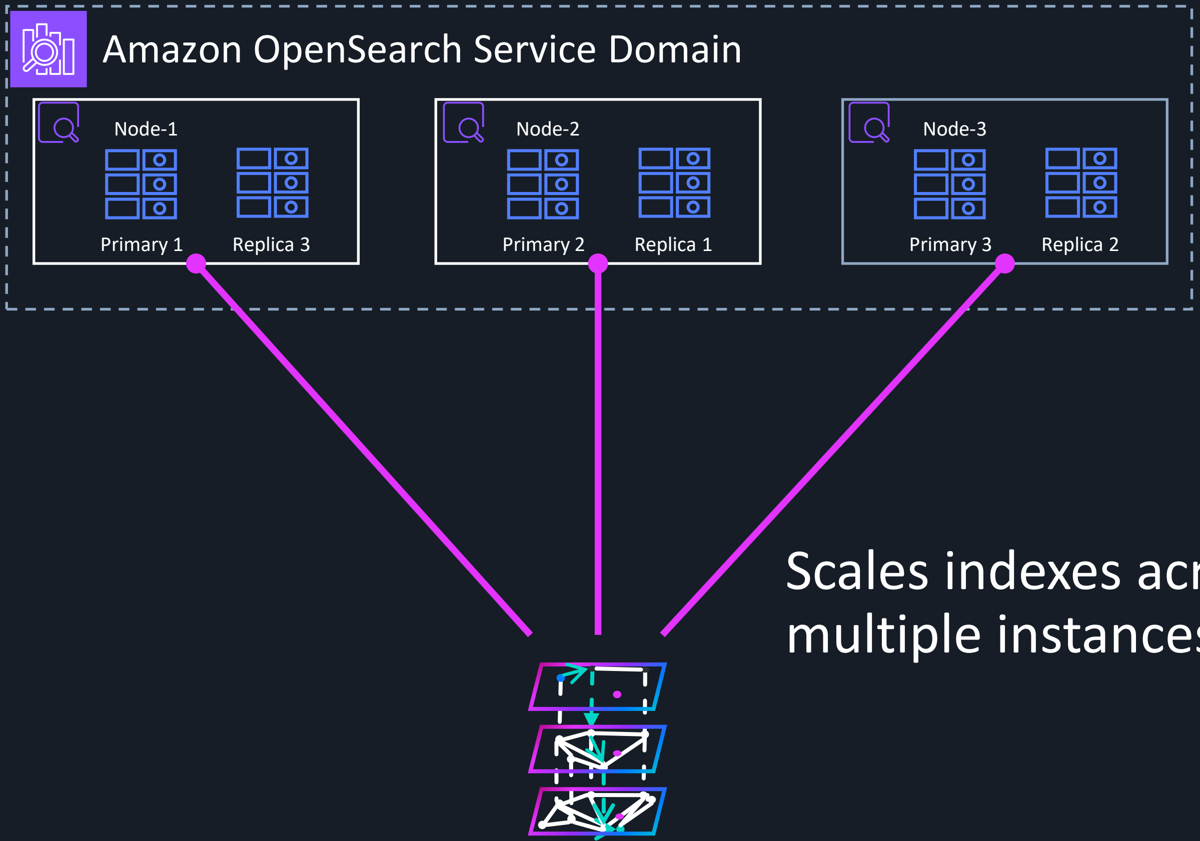
Approximated k-NN

Hierarchical Navigable Small Worlds (HNSW)



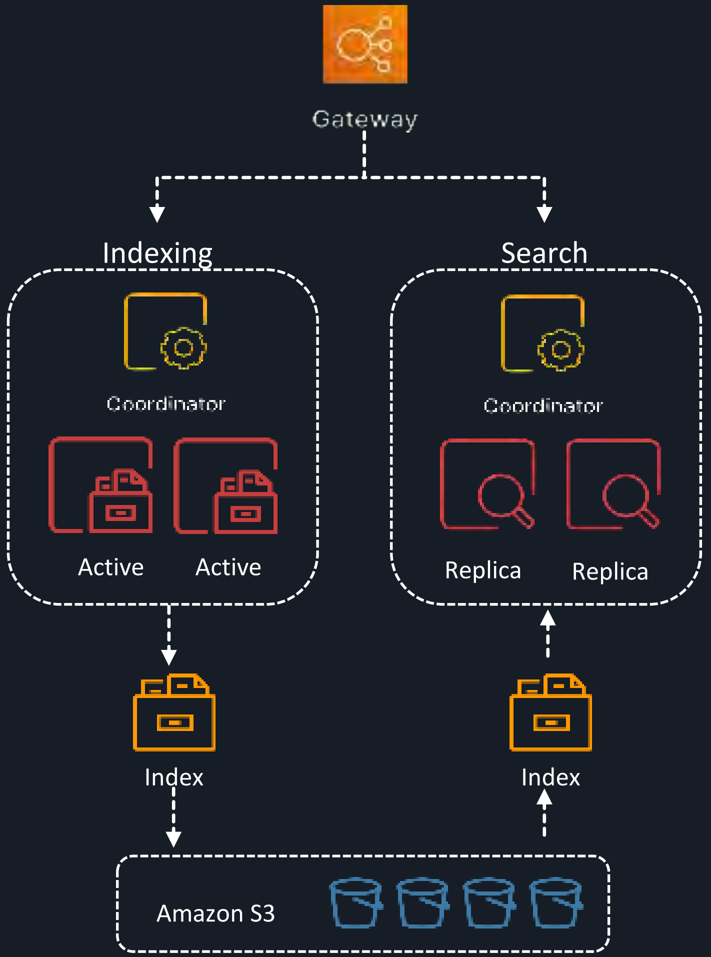
Scaling Vector Search

Managed Clusters



Scales indexes across multiple instances

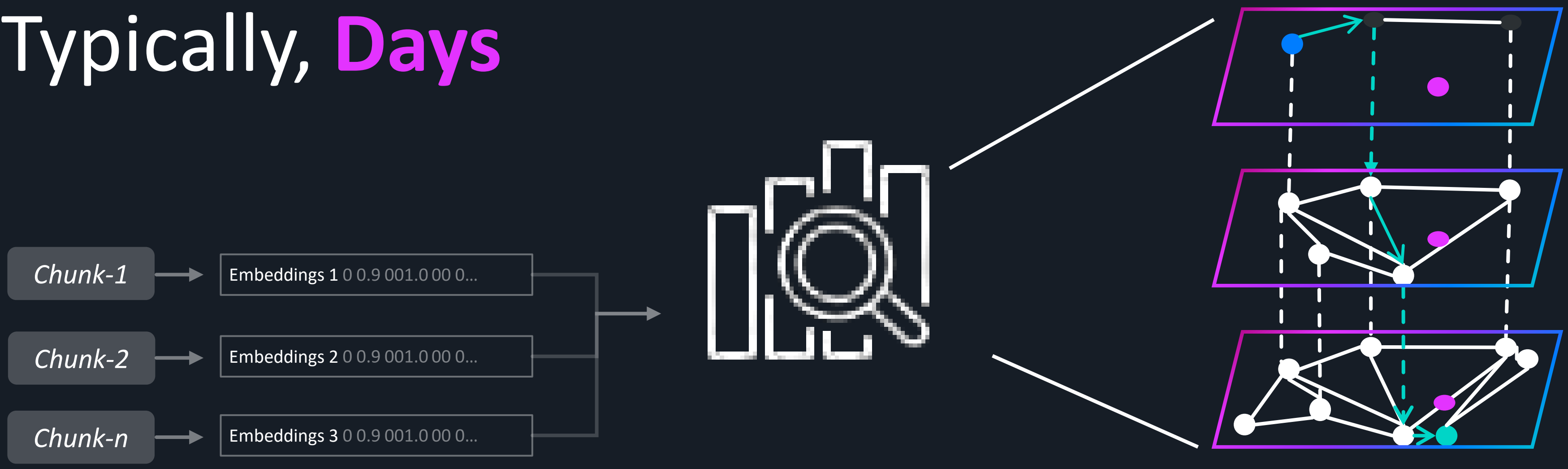
Serverless Collections



How long does it take to index 1-Billion?

Build Index

Typically, **Days**

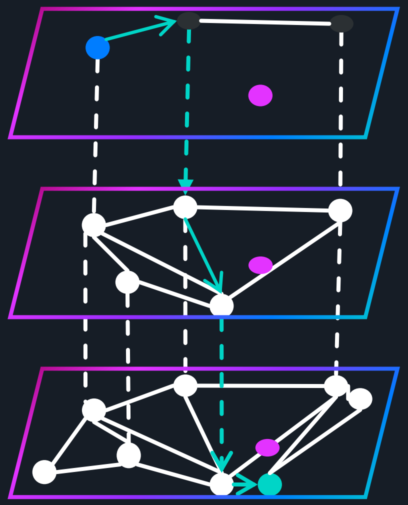


1-Billion Vectors

OpenSearch Service Vector Database

Build indexes

Life-cycle of a vector index…



Data Changes

Add, modify and delete content

(index (HNSW) recall degradation)

Model Changes

New provider or version

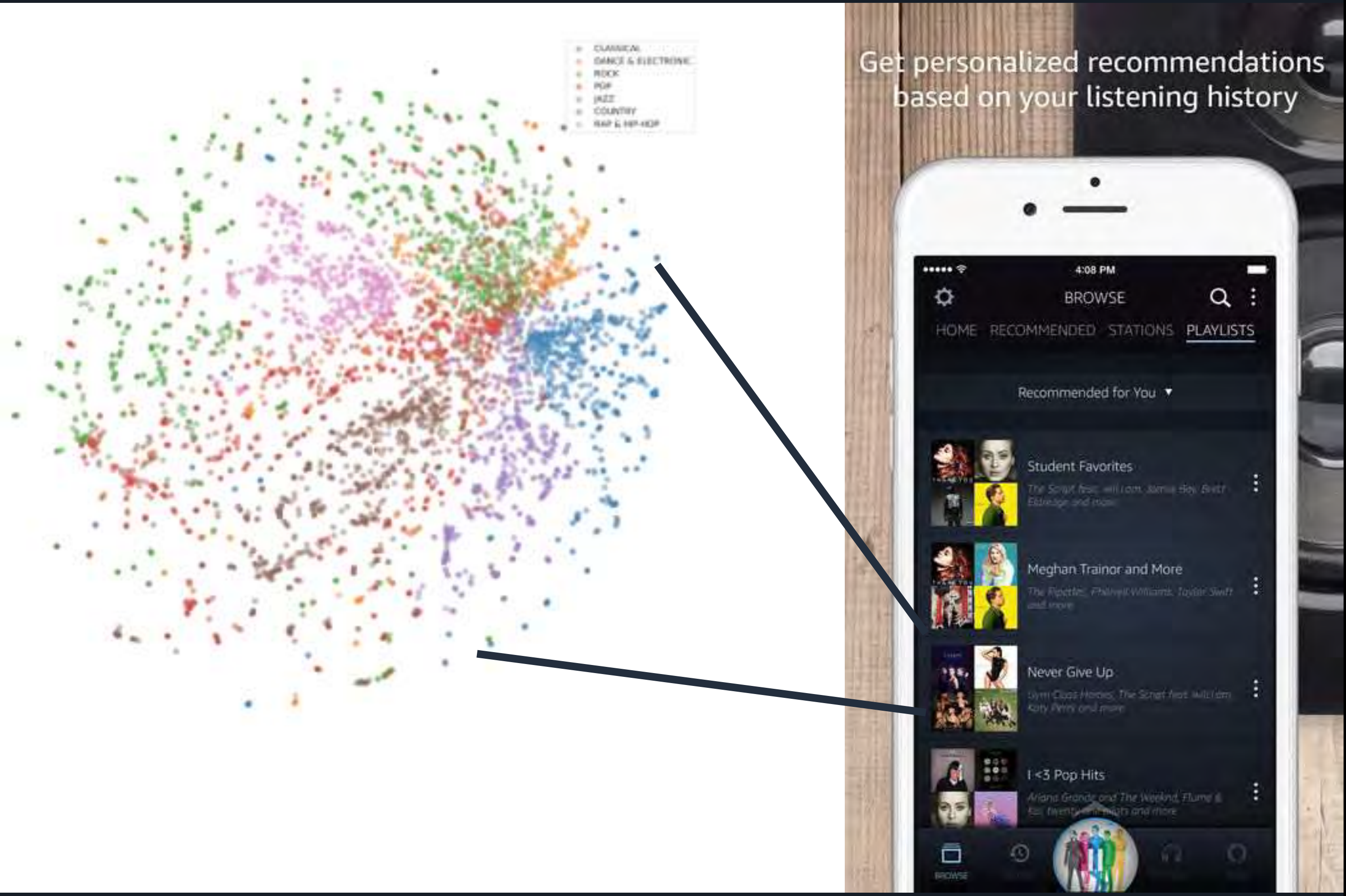
Model Changes

Fine-tuning

Personalization

amazon music

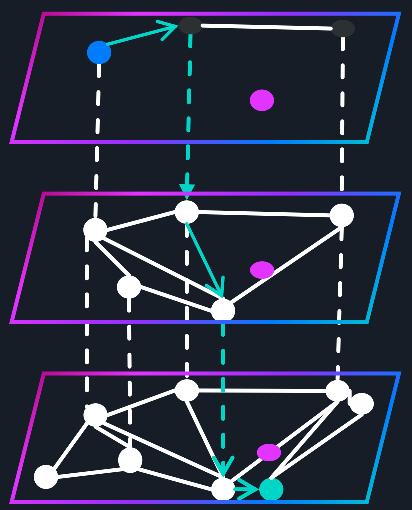
- **100 Million** music tracks with recommendations based on user listening history.
- **1.05 Billion** vectors indexed into OpenSearch to power item-item collaborative filtering.
- **Daily** model retraining to deliver high-quality recommendations



Source *https://aws.amazon.com/blogs/big-data/amazon-opensearch-services-vector-database-capabilities-explained/*

How to maintain fast search on dynamic applications?

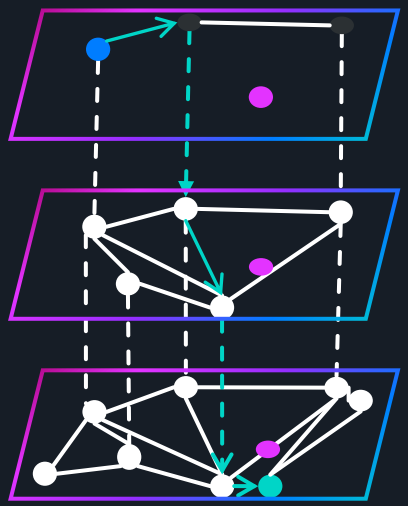
Compete for significant compute and RAM



Indexing

Bulk, reindex, update, delete, index, force-merge, flush…

OpenSearch
Service
Managed Cluster
(Domain)



Search

Search.query.knn

Challenges at scale...

- Index build and maintenance takes days.
- Vector ingestion can impact search times.

amazon music

DevRev

Help customers...

- Maintain innovation velocity and productivity
- Build responsive, dynamic, AI applications

GPU-accelerated Vector Indexing

NVIDIA cuVS

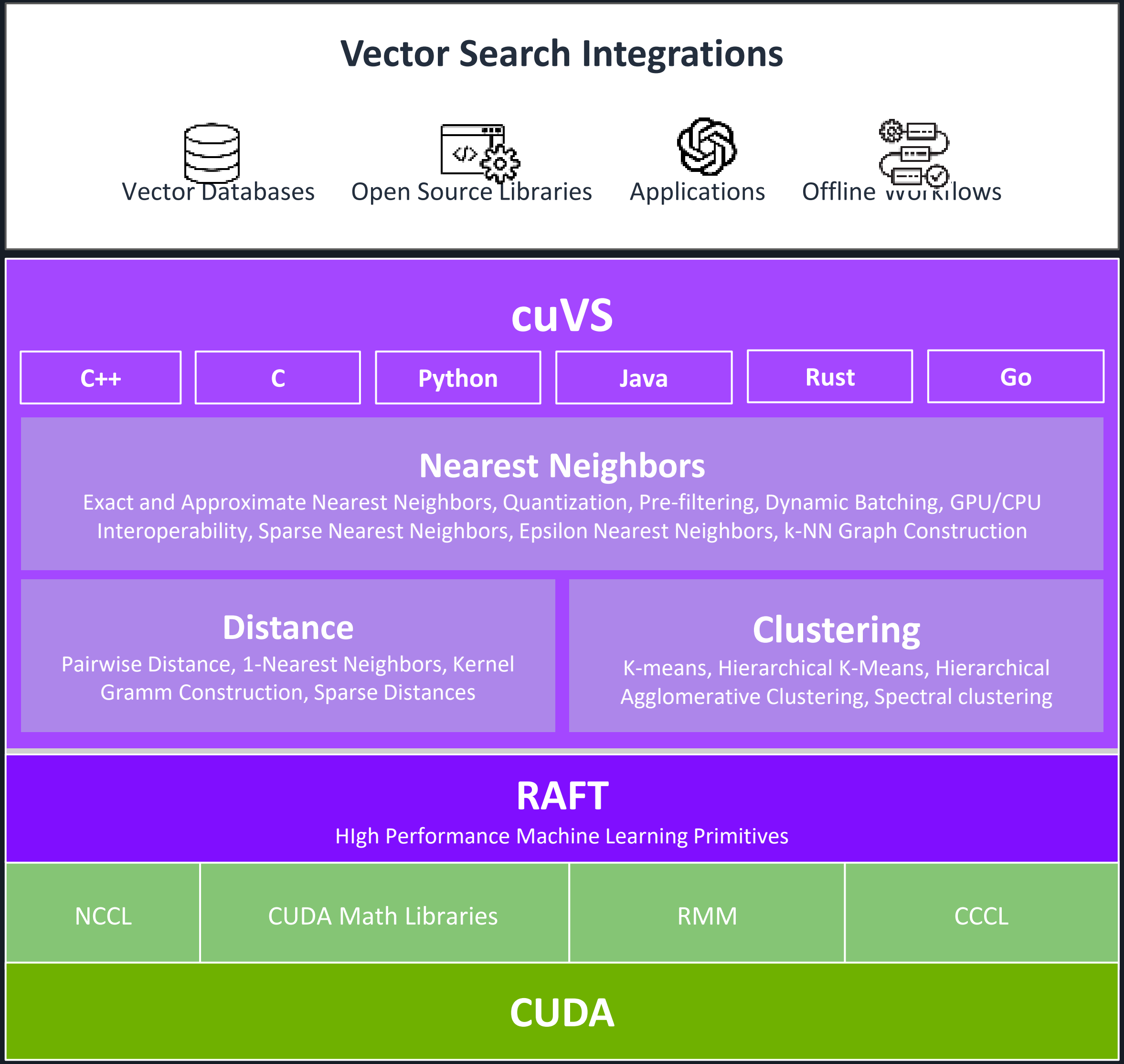
Best Performance
20X faster index build time, 11X lower latency

Advanced Algorithms
Performance-tuned approximate nearest neighbor search

Flexible Integration
Supports multiple languages including C, C++, Python, and Rust

Interoperable
interoperable between CPU and GPU

Scalable
Enables massive-scale vector search and clustering



https://developer.nvidia.com/cuvs

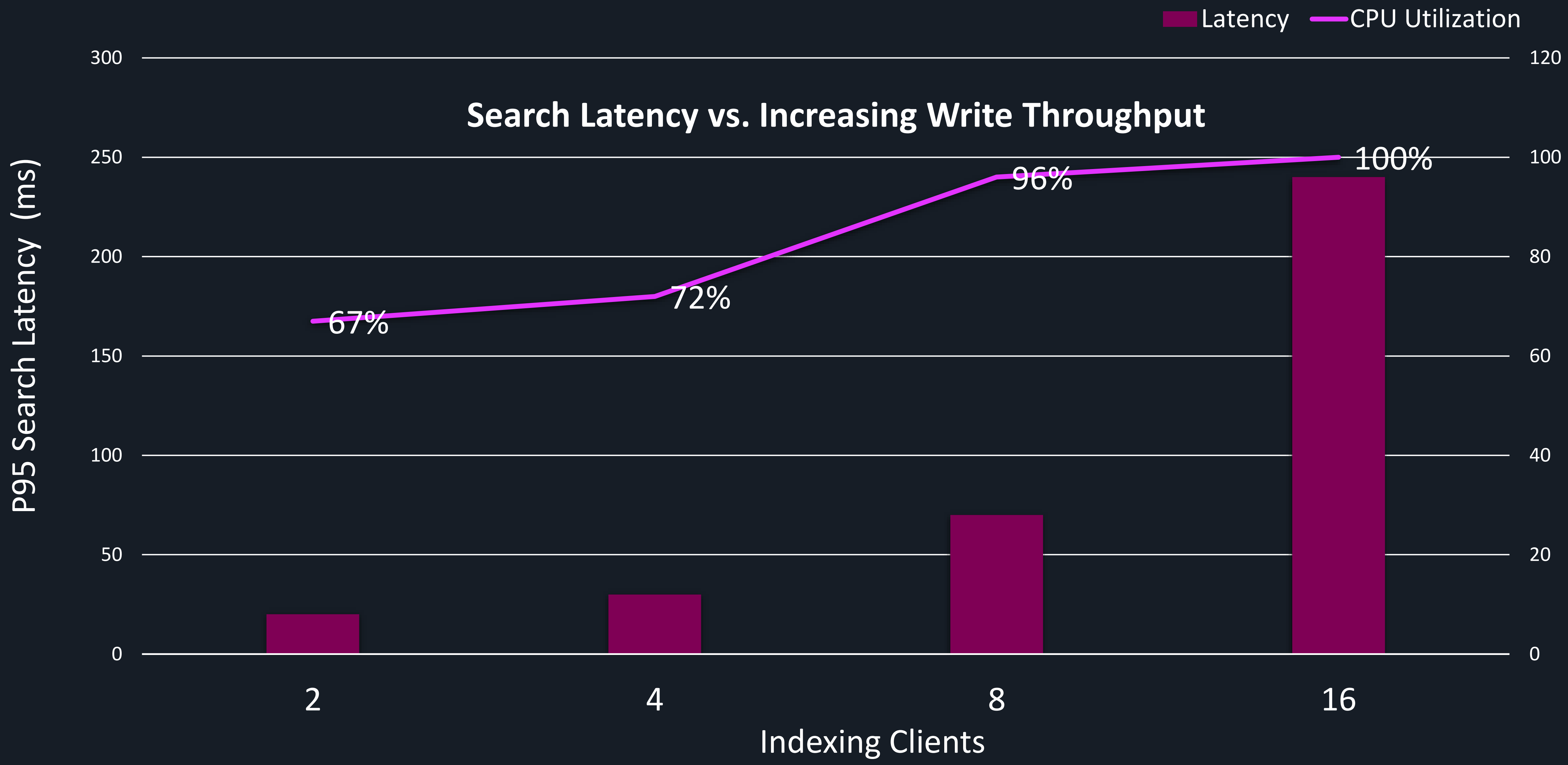
GPU-Acceleration on Managed Clusters (Domains)

Vectors	CPU-only (Index + Merge)	Add GPU	Speed Gain	Compute Cost
1M 768-dim	1.4 hr.	9.9 min.	8X	8X Less
10M 768-dim	8.5 hr.	36.8 min.	14X	12X Less
113M 1024-dim	28.7 hr.	4.5 hr.	6X	6X Less
1B 128-dim	31.9 hr.	2.8 hr. **(Index: 35 min.)**	11X	10X Less

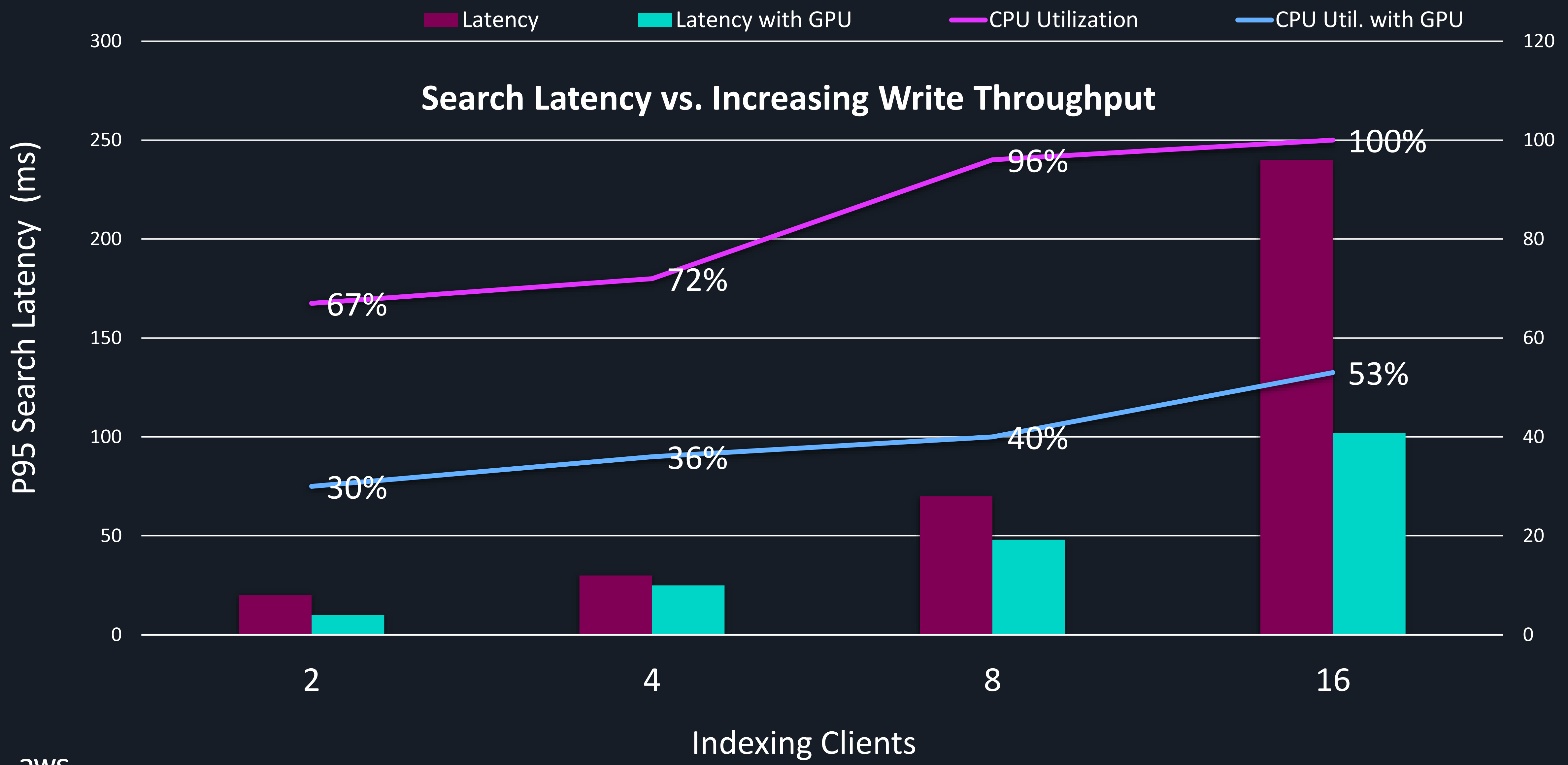
GPU-Acceleration on Serverless (Collections)

Vectors	Index OCU/hrs.	With GPU	Cost Reduction
1M 768-dim	8	1.5	5.3X
10M 768-dim (min. 32 OCUs)	78	20.3	3.8X
113M 1024-dim (min. 48 OCUs)	2721	304.5	8.9X **$653 vs. $73**
1B 128-dim (min. 48 OCUs)	1562	201	7.8X **$375 vs. $48**

Indexing Impacts Search Speed on Managed Clusters



Offloading Indexing to GPU Improves Search Speed



How can we deliver practical economics?

CPU Cluster

3 X 384 RAM

r8g.12xlarge.search:
48 vCPU, 384 RAM

1B 1024 dim Vectors
32X compression

Indexing < 30%
of uptime

GPU Cluster

6 X 192 RAM

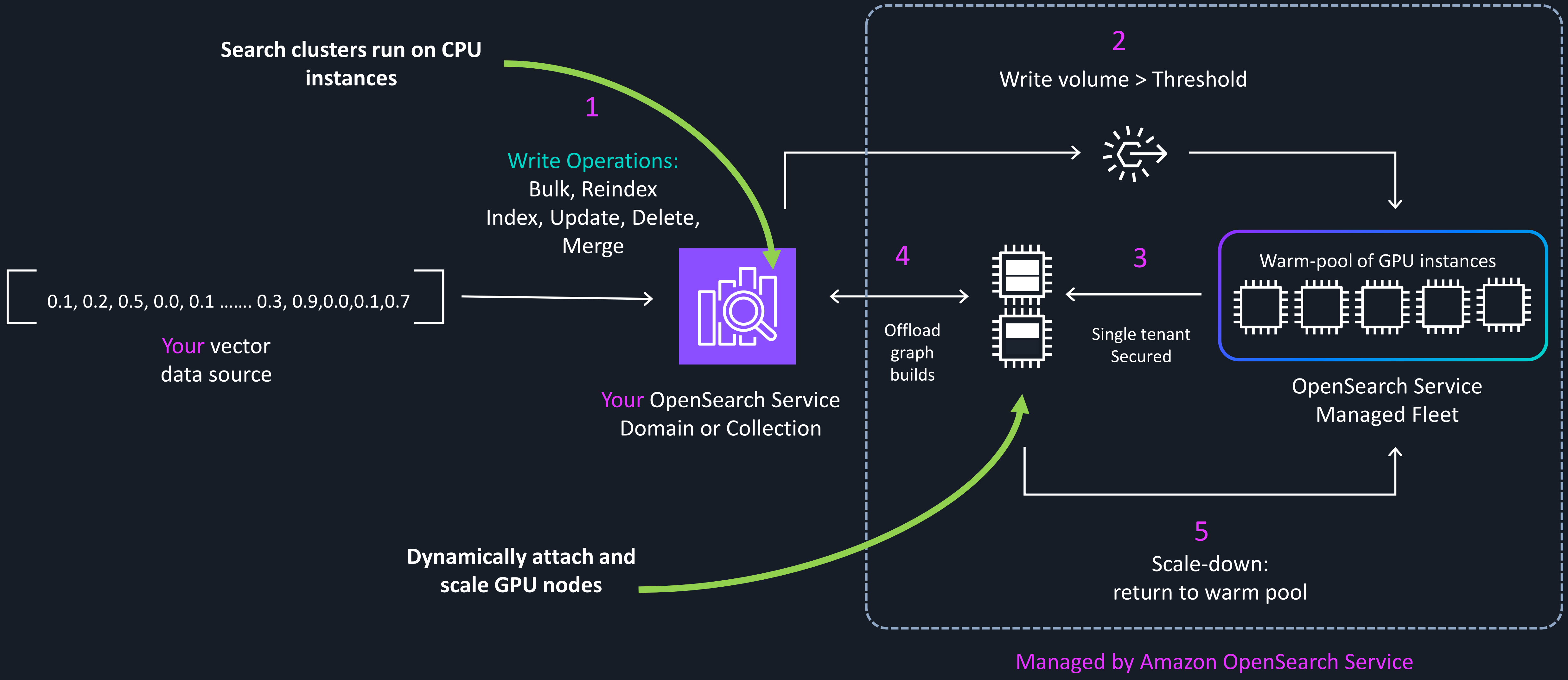
g6.12xlarge.search:
48 vCPU, 192 RAM

2.4X Cost

Overprovisioned GPU

Wasted $ from poor utilization

On-demand GPU acceleration, pay-for-value



Serverless Acceleration for Domains and Collections

1

Enable on
Domain or Collection

2

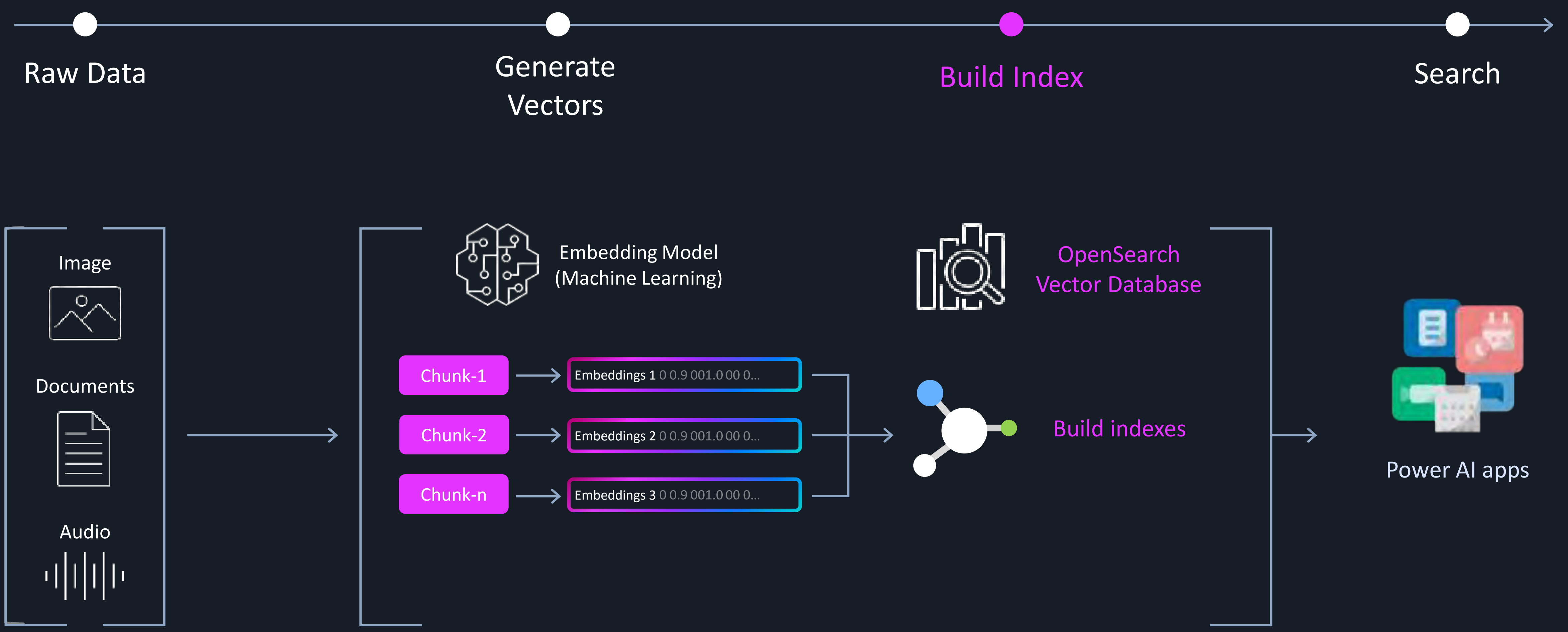
Pay-on-use

OpenSearch Compute Unit
(OCU)—Vector Acceleration

$0.24 OCU/hr. (N. Virginia)

Optimizing Vector Indexes

Building a Vector Database (Recap...)



Building a Vector Database

Raw Data → Generate Vectors → **Optimize Index Configurations** → **Build Index**

Image

Documents

Audio

Embedding Model (Machine Learning)

Chunk-1 → Embeddings 1 0 0.9 001.0 00 0...

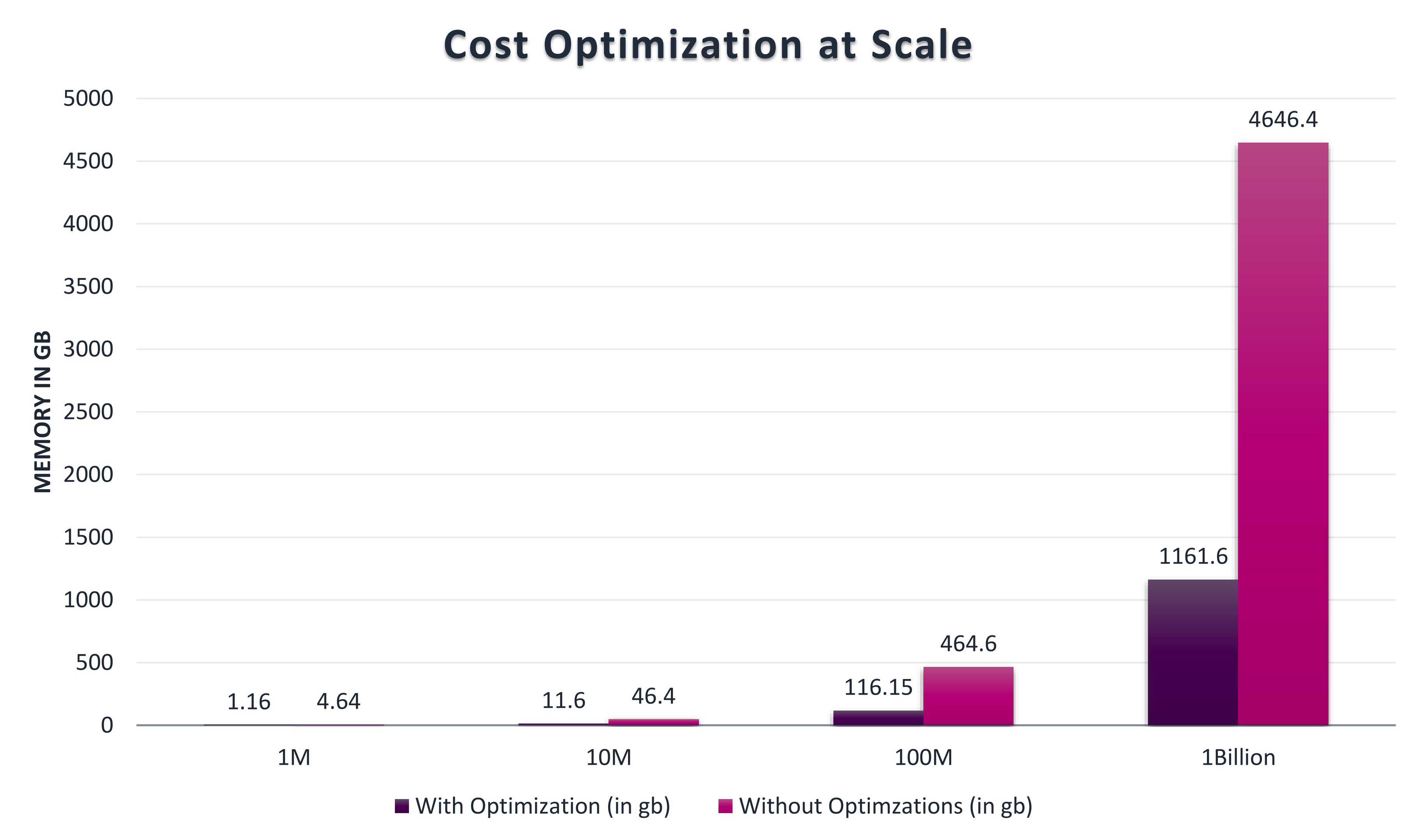
Chunk-2 → Embeddings 2 0 0.9 001.0 00 0...

Chunk-n → Embeddings 3 0 0.9 001.0 00 0...

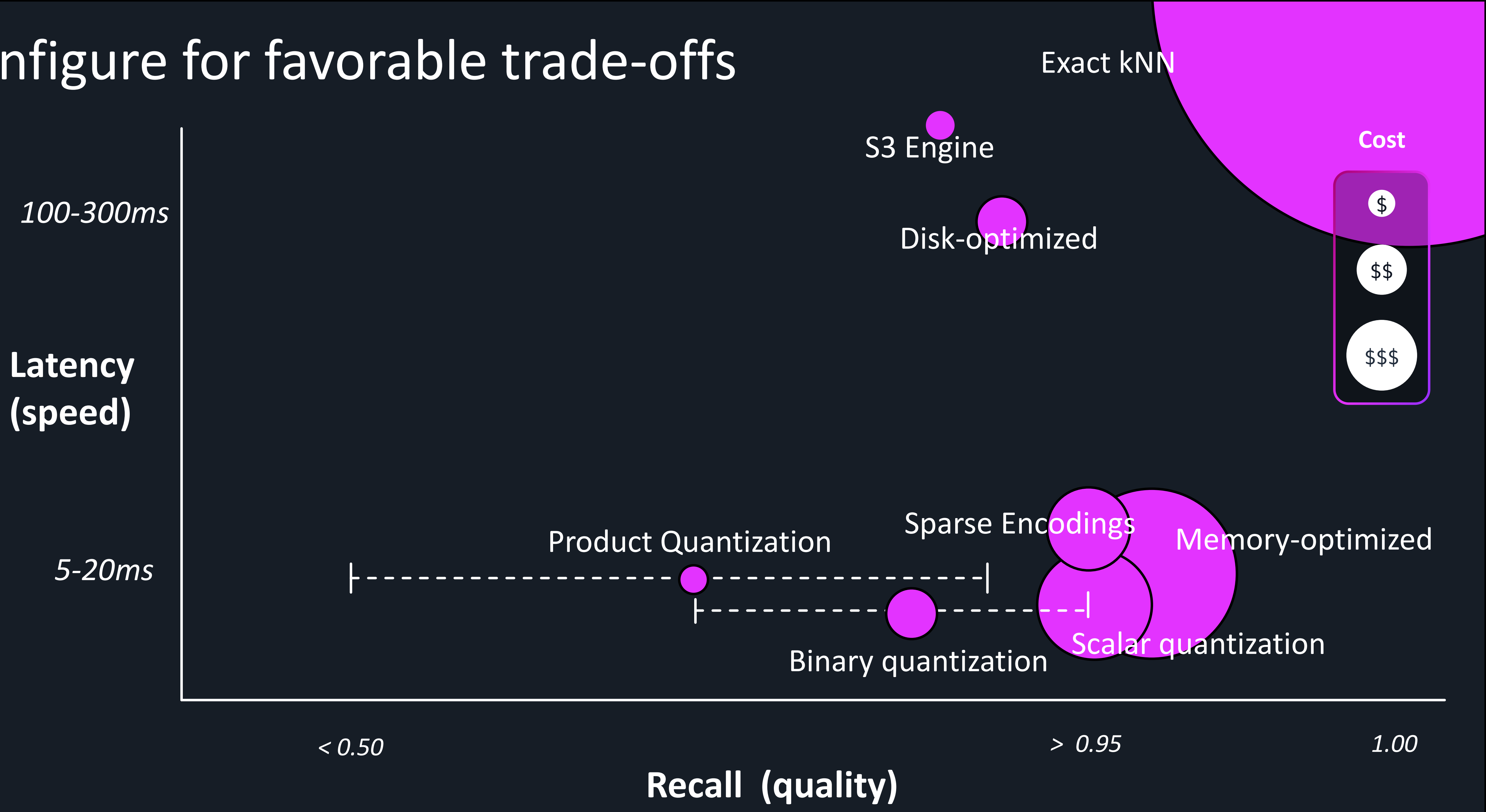
OpenSearch Vector Database

Build indexes

Cost to host a 1-Billion vector index



Configure for favorable trade-offs



Time-consuming, expert-driven process

Select Index Parameters

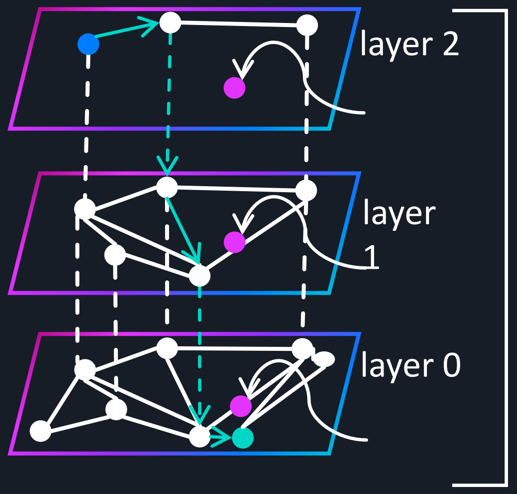
Algorithms:
HNSW, ef_construction, m…

Quantization:
Scalar, Binary, Product

Engine Settings:
Disk-optimized,
 In-memory,
Infrequent Queries

Build and Evaluate Index

1 Search latency

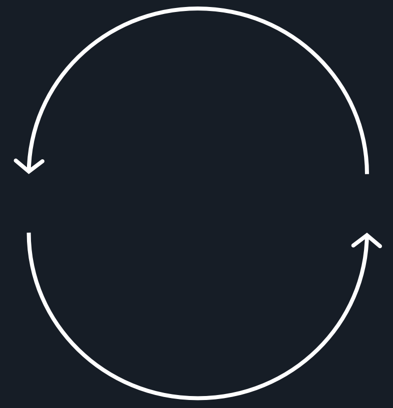


2 Memory Footprint

3 Recall

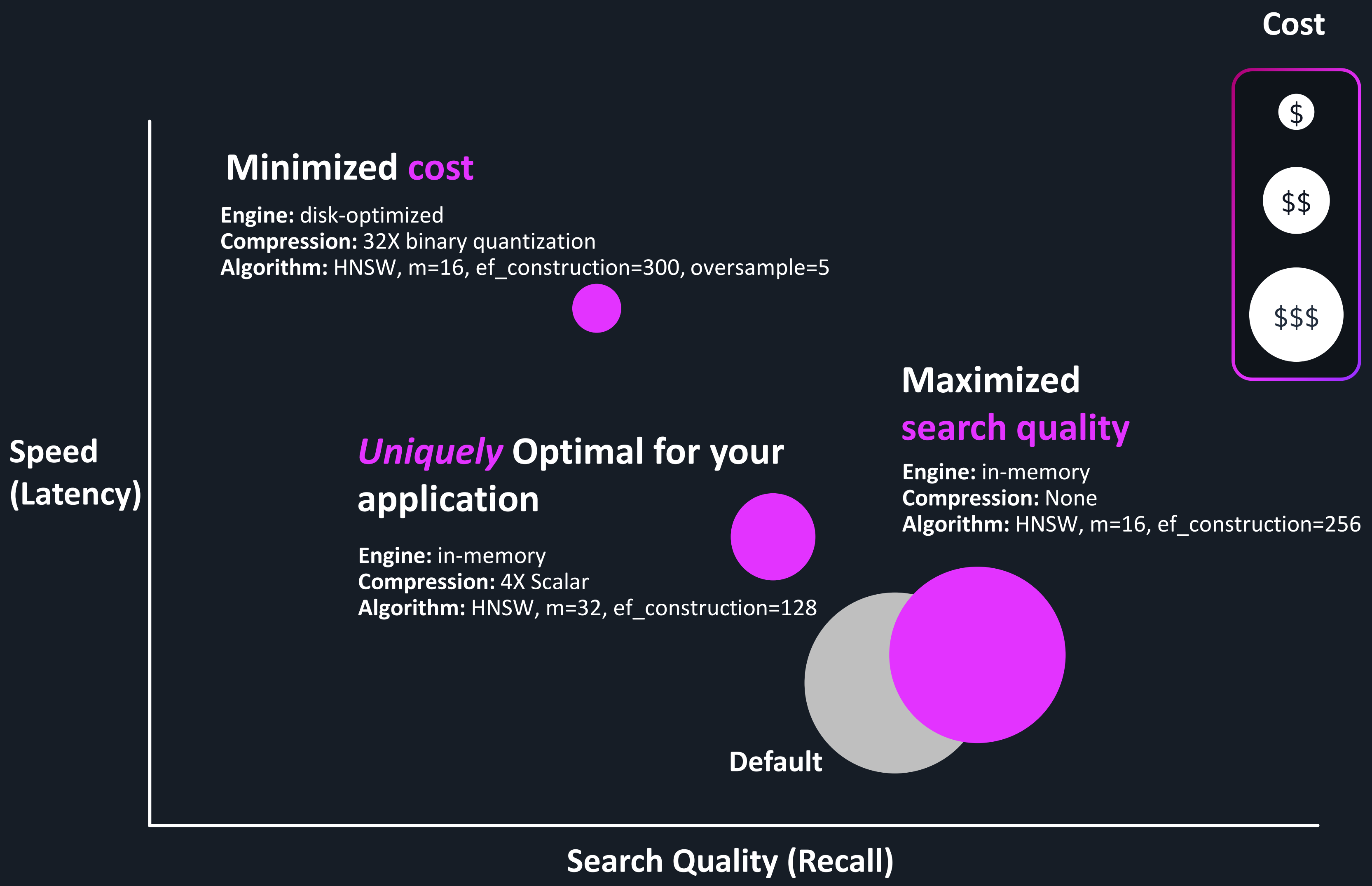
Exact (Brute-force) k-NN

Adjust Parameters and Repeat

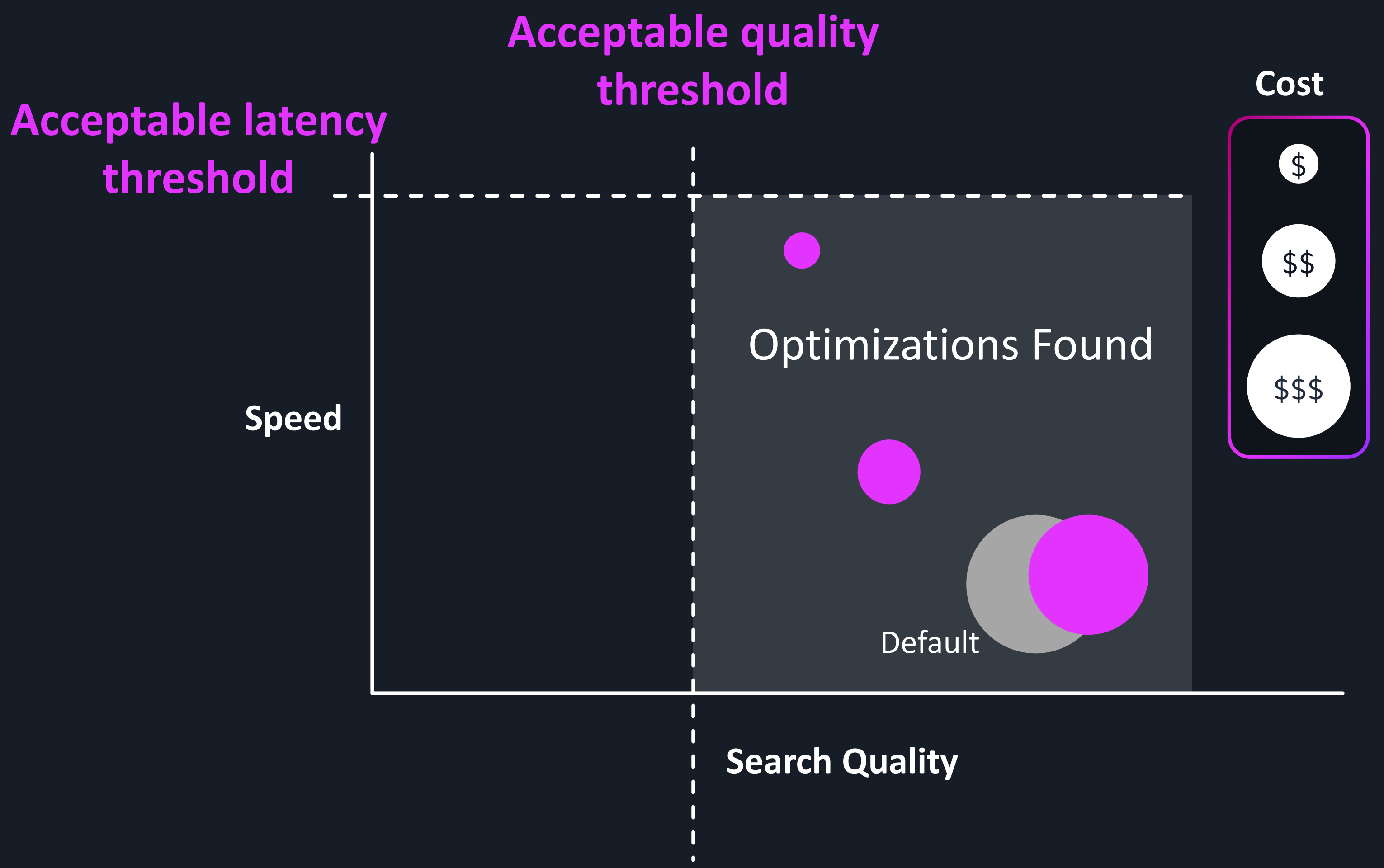


aws

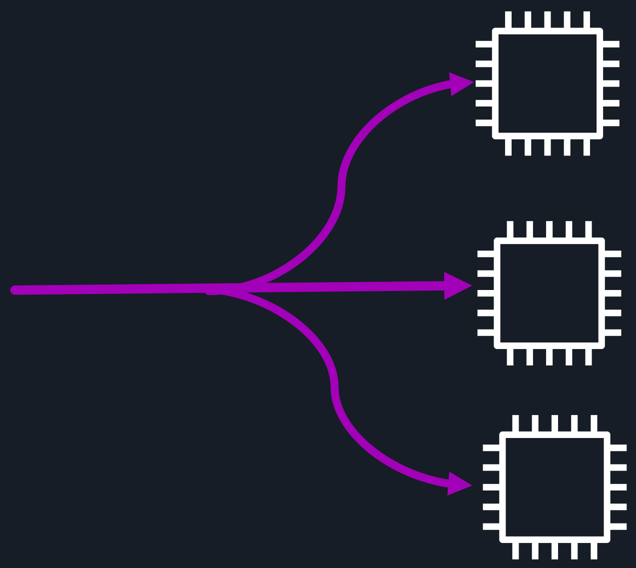
Best optimizations depend on data and use case



Let’s simplify! Auto-optimize Vectors



Parallelize Index Builds and Evaluations



Serverless Auto-optimize jobs with a predictable flat rate

Demo: Build an auto-optimized, GPU-accelerated vector database

Level up your skills on AWS Skill Builder

Build AWS Cloud and AI skills, your way, with our online learning center.

LEARN
1,000+ free learning resources

PRACTICE
Hands-on labs and immersive real-world simulations powered by AI

PREPARE
Official AWS Certification exam prep

VALIDATE
Microcredentials to validate practical skills

Scan to start learning

skillbuilder.aws

Thank you

Please complete the session survey in the mobile app