zilliz

Vector Databases 101

Stephen Batifol | Zilliz

Conf42 - Machine Learning Track

Product Portfolio

Open Source

milvus
VECTOR DATABASE

Knowhere
VECTOR SEARCH ENGINE

GPT-Cache
SEMANTIC CACHE FOR LLM QUERIES

VDB Benchmark
VECTORDB BENCHMARK TOOL

Attu
GUI for Milvus

Commercial Offerings

Zilliz Cloud

Optimized Milvus with essential data and security tools for a high-performing vector search platform

Deploy fully managed or "Bring Your Own Cloud" (BYOC)

aws

Microsoft Azure

zilliz

Partner with Industry Leaders

Cloud Service Provider	aws	Google Cloud	Azure
Data Platform	databricks	CONFLUENT	
GenAI Tooling	cohere	LlamaIndex	LangChain
Chip Manufacturer	NVIDIA	intel	

zilliz

CONTENTS

zilliz

01

Why Vector Databases?

Why Vector Databases?

- Unstructured Data is 80% of data

- Vector Databases are really good with unstructured data

- Examples of Unstructured Data include text, images, videos, audio, etc

zilliz

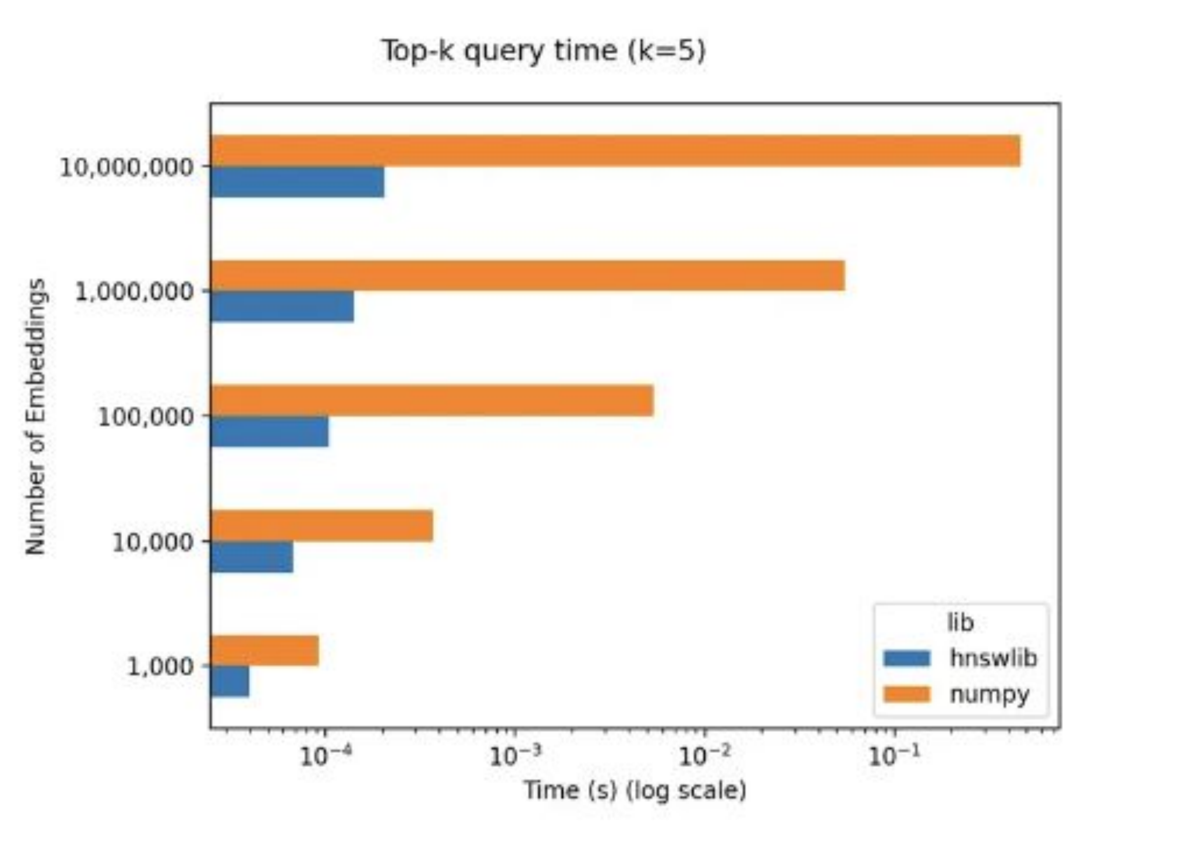
As Easy as a Numpy KNN?

```python
import numpy as np

# Function to calculate Euclidean distance
def euclidean_distance(a, b):
    return np.linalg.norm(a - b)

# Function to perform KNN
def knn(data, target, k):
    # Calculate distances between target and all points in data
    distances = [euclidean_distance(d, target) for d in data]
    # Combine distances with data indices
    distances = np.array(list(zip(distances,
range(len(data)))))
    # Sort by distance
    sorted_distances = distances[distances[:, 0].argsort()]
    # Get the top k closest indices
    closest_k_indices = sorted_distances[:k, 1].astype(int)
    # Return the top k closest vectors
    return data[closest_k_indices]
```

Scale is a problem



Why Not Vector Search Libraries?

- Search Quality - Hybrid Search? Filtering?
- Scalability - Billions of vectors?
- Multi tenancy - Isolating Multi-Tenant data
- Cost - Memory, disk, S3?
- Security - Data Safety and Privacy

TL;DR: Vector search libraries lack the infrastructure to help you scale, deploy, and manage your apps in production.

 |

zilliz

Why Not Use a SQL/NoSQL Database?

- Inefficiency in High-dimensional spaces
- Suboptimal Indexing
- Inadequate query support
- Lack of scalability
- Limited analytics capabilities
- Data conversion issues

TL;DR: Vector operations are too computationally intensive for traditional database infrastructures

What is Milvus/Zilliz ideal for?

Purpose-built to store, index and query vector embeddings from unstructured data **at scale**.

- Advanced filtering
- Hybrid search
- Multi-vector Search
- Durability and backups
- Replications/High Availability
- Sharding
- Aggregations
- Lifecycle management
- Multi-tenancy

- High query load
- High insertion/deletion
- Full precision/recall
- Accelerator support (GPU, FPGA)
- Billion-scale storage

 |

zilliz

Takeaway:

Vector Databases are **purpose-built** to handle **indexing, storing, and querying** vector data.

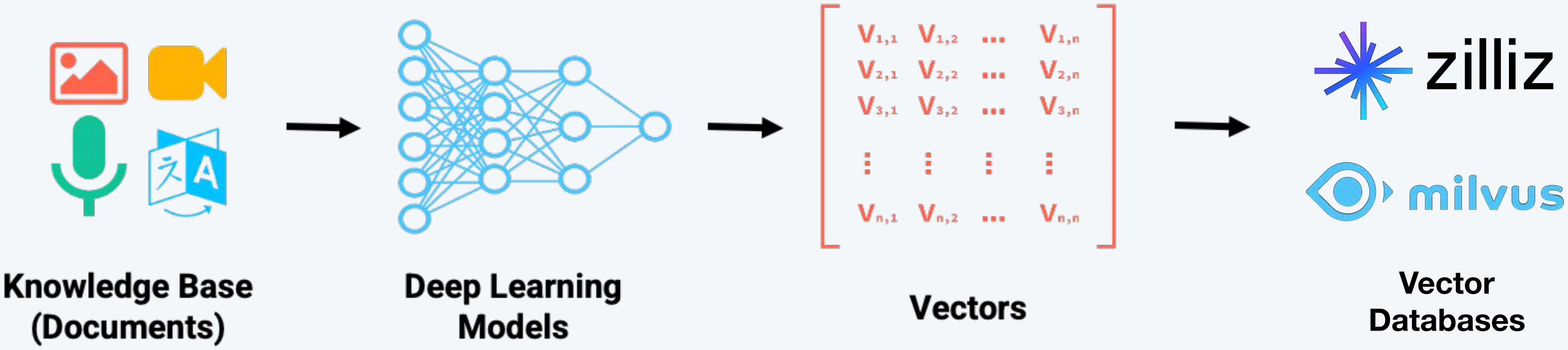
Milvus & Zilliz are specifically designed for high performance and **billion+** scale use cases.

zilliz

02

Where do vectors come from?

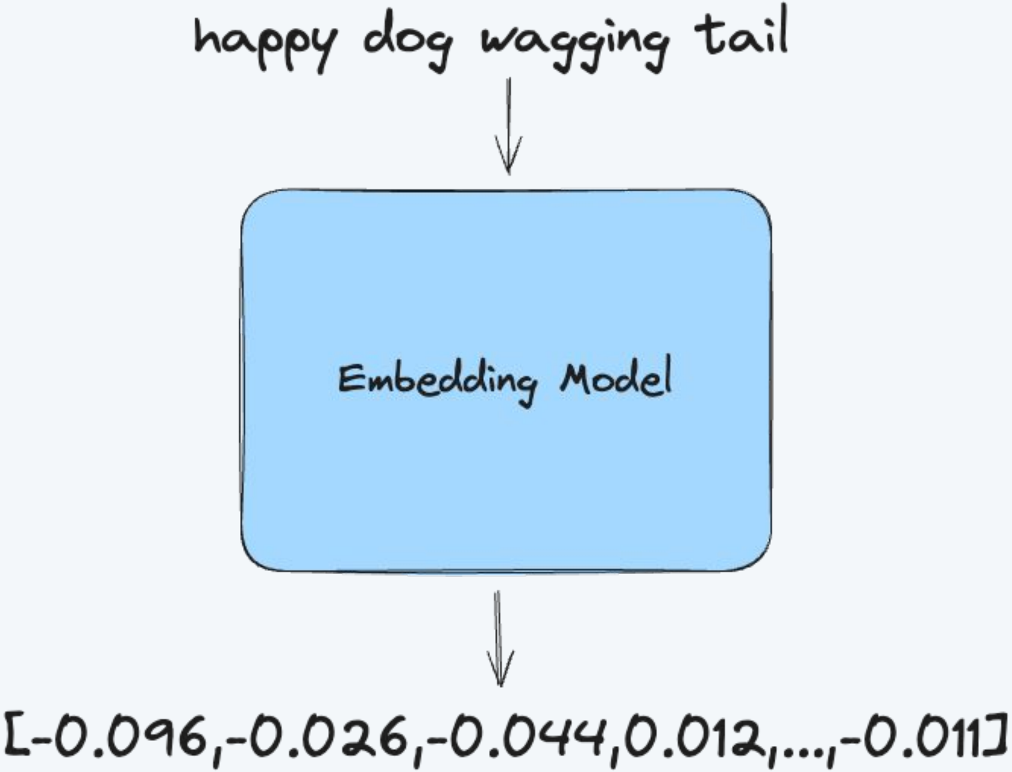
Where do Vectors Come From?



 |

Embeddings Models

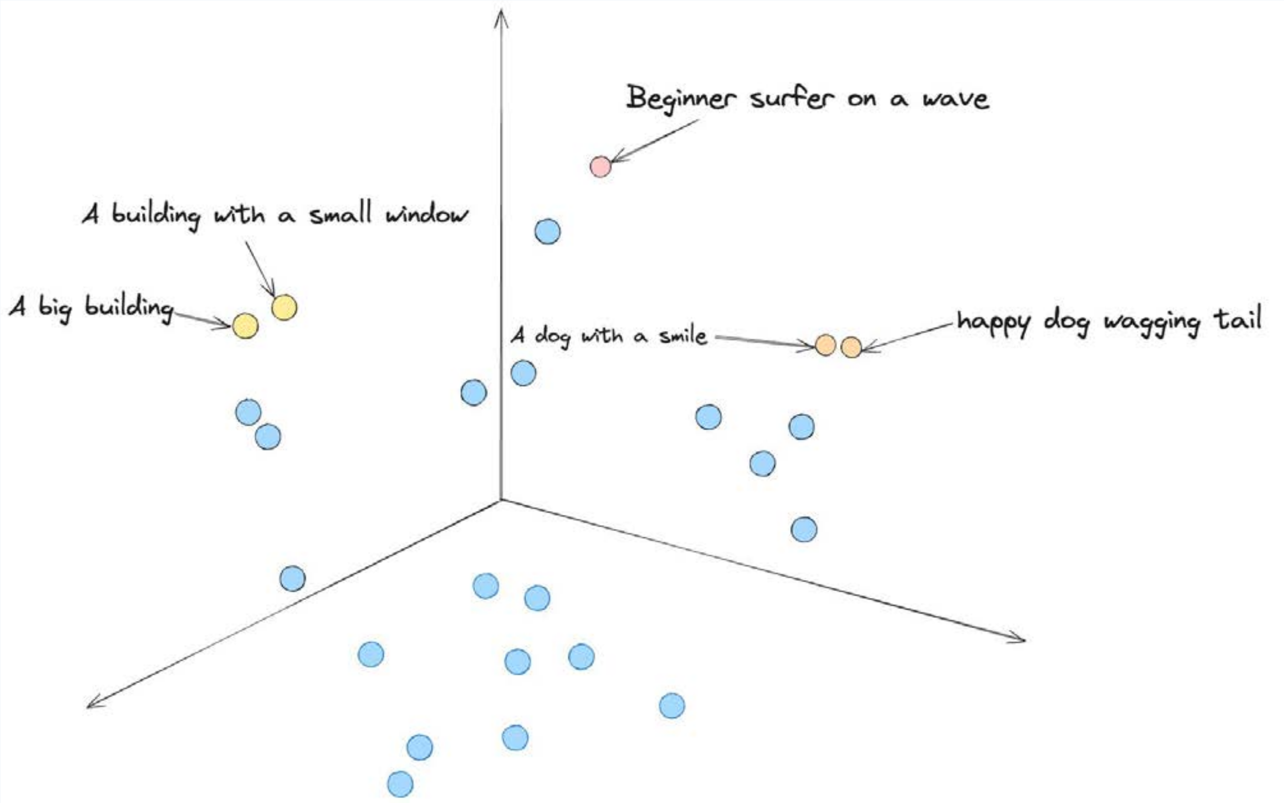
Vector Embedding



 |

zilliz

Vector Space



 |

03

Vector Database Use Cases

zilliz

Common AI Use Cases

Retrieval Augmented Generation (RAG)

Expand LLMs' knowledge by incorporating external data sources into LLMs and your AI applications.

Recommender System

Match user behavior or content features with other similar ones to make effective recommendations.

Text/ Semantic Search

Search for semantically similar texts across vast amounts of natural language documents.

Image Similarity Search

Identify and search for visually similar images or objects from a vast collection of image libraries.

Video Similarity Search

Search for similar videos, scenes, or objects from extensive collections of video libraries.

Audio Similarity Search

Find similar audios in large datasets for tasks like genre classification or speech recognition

Molecular Similarity Search

Search for similar substructures, superstructures, and other structures for a specific molecule.

Anomaly Detection

Detect data points, events, and observations that deviate significantly from the usual pattern

Multimodal Similarity Search

Search over multiple types of data simultaneously, e.g. text and images

04

How do Vector Databases Work?

zilliz

Example Entry

"id": "https://towardsdatascience.com/detection-of-credit-card-fraud-with-an-autoencoder-927585
"embedding": [-0.042092223,-0.0154002765,-0.014588429,-0.031147376,0.03801204,0.013369046,
"date": "2023-06-01"
"paragraph": "We define an anomaly as follows:"
"reading_time": "11"
"subtitle": "A guide for the implementation of an anomaly..."
"publication": "Towards Data Science"
"responses": "1"
"article_url": "https://towardsdatascience.com/detection-of-credit-card-fraud-with-an-autoencoder-
"title": "Detection of Credit Card Fraud with an Autoencoder"
"claps": "229"

Hide 6 fields

Vector search

05

What is Similarity Search?

zilliz

Vector Search Overview

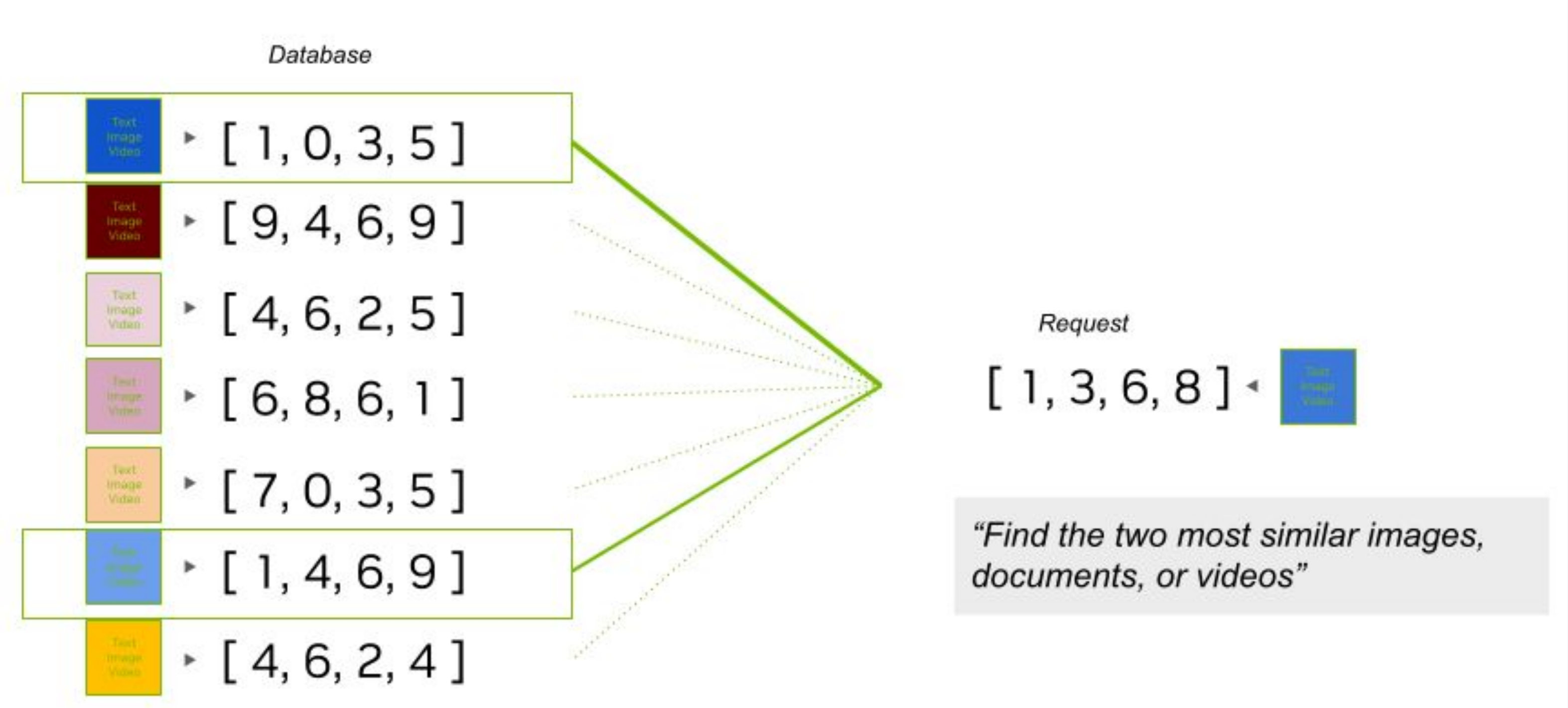
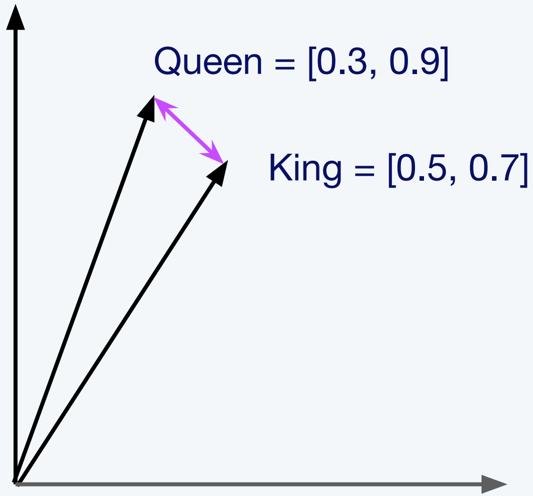


Image from Nvidia

Vector Similarity Measures: L2 (Euclidean)

$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$



$$\begin{aligned} d(\text{Queen, King}) &= \sqrt{(0.3-0.5)^2 + (0.9-0.7)^2} \\ &= \sqrt{(0.2)^2 + (0.2)^2} \\ &= \sqrt{0.04 + 0.04} \\ &= \sqrt{0.08} \cong 0.28 \end{aligned}$$

 |

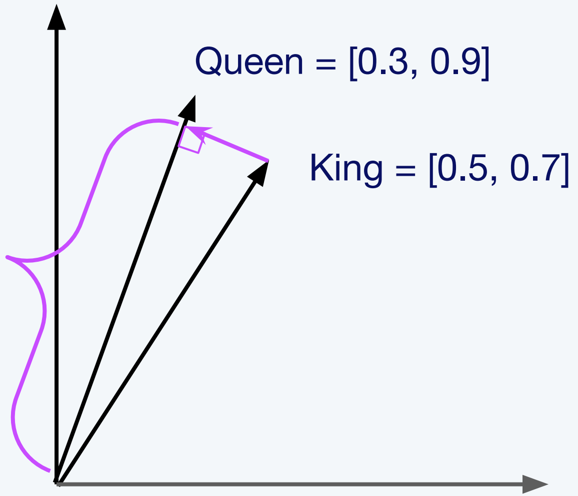
zilliz

Vector Similarity Measures: Inner Product (IP)

$$a \cdot b = \sum_{i=1}^{n} a_i b_i$$

Queen · King = (0.3*0.5) + (0.9*0.7)

= 0.15 + 0.63 = 0.78



Vector Similarity Measures: Cosine

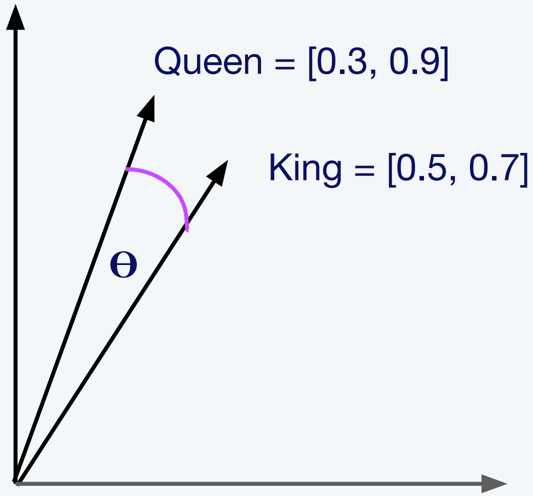
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

$$\text{cos(Queen, King)} = \frac{(0.3*0.5)+(0.9*0.7)}{\sqrt{0.3^2+0.9^2} * \sqrt{0.5^2+0.7^2}}$$

$$= \frac{0.15+0.63}{\sqrt{0.9} * \sqrt{0.74}}$$

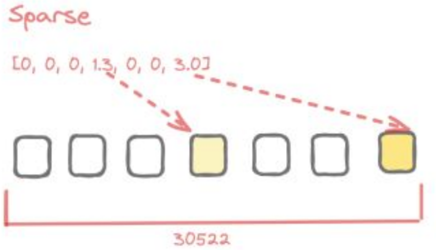
$$= \frac{0.78}{\sqrt{0.666}}$$

$$\cong 0.03$$



zilliz

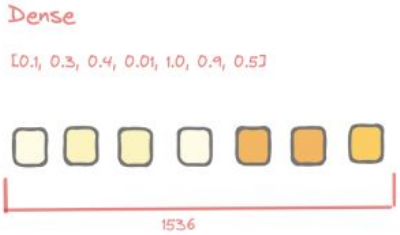
Choosing Vector Embedding Types



Distance： IP

Models: Splade, BGE-M3

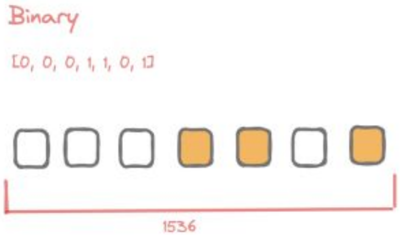
Index: Wand, Graph



Distance： IP, L2, Cosine

Models: OpenAI, BGE, Cohere

Index: Faiss, HNSW



Distance： Hamming, Superstructure, Jaccard, Tanimoto

Models: Cohere, Meta ESM-2

Index: Faiss

zilliz

06

Indexes

Indexing strategies

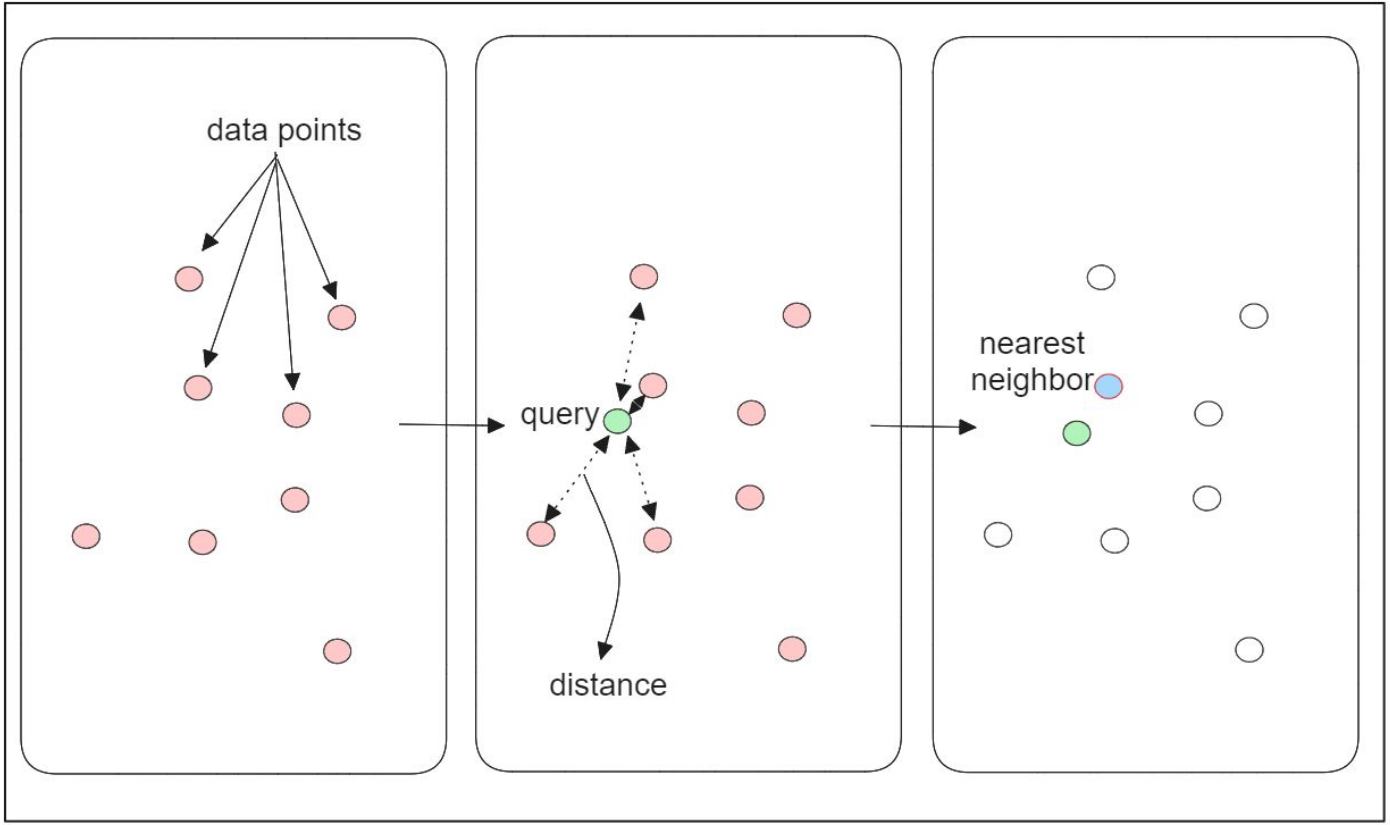
- Tree based
- Graph based
- Hash based
- Cluster based

Category	Index	Accuracy	Latency	Throughput	Index Time	Cost
Graph-based	Cagra (GPU)	High	Low	Very High	Fast	Very High
	HNSW	High	Low	High	Slow	High
	DiskANN	High	High	Mid	Very Slow	Low
Quantization-based or cluster-based	ScaNN	Mid	Mid	High	Mid	Mid
	IVF_FLAT	Mid	Mid	Low	Fast	Mid
	IVF + Quantization	Low	Mid	Mid	Mid	Low

zilliz

FLAT

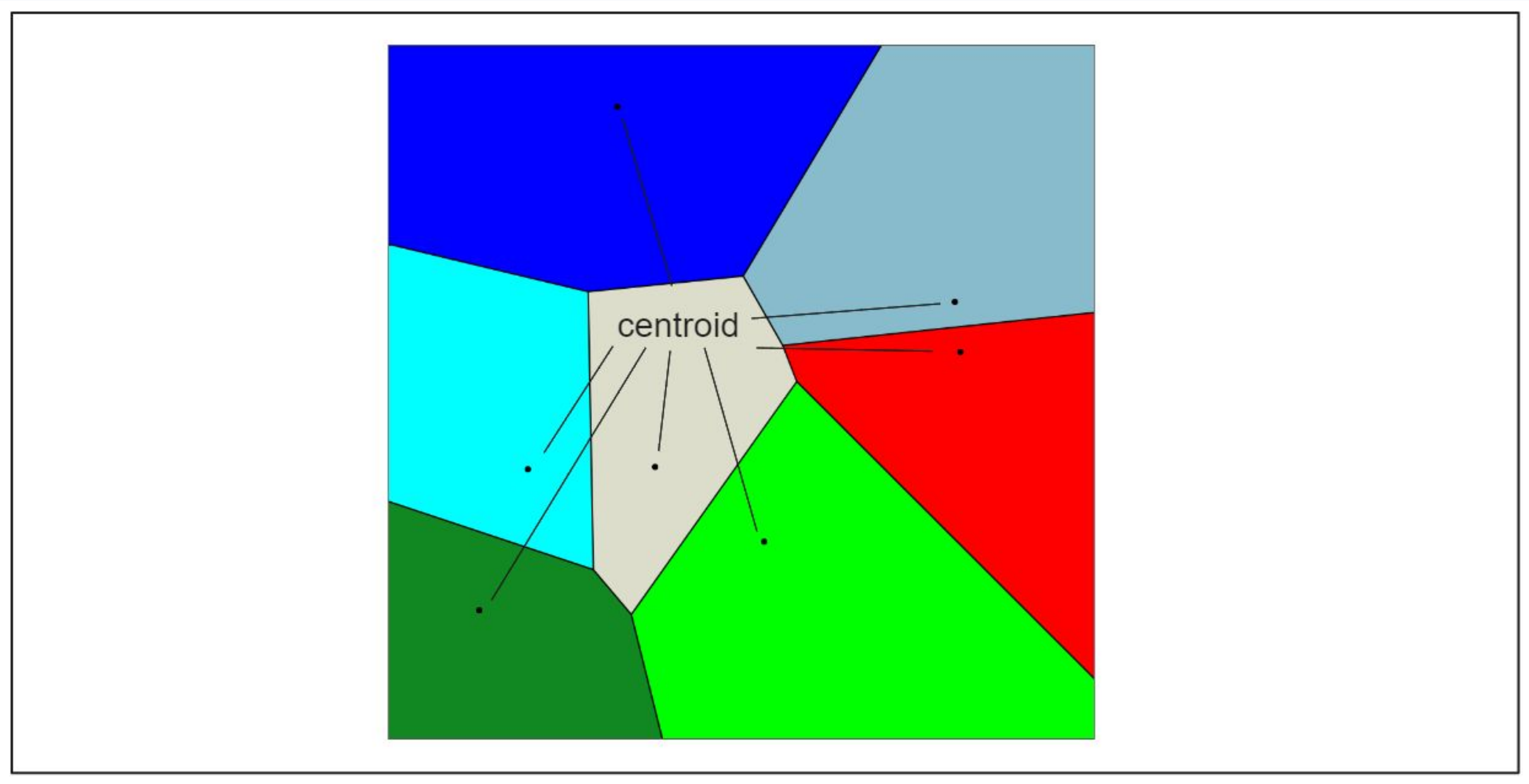
FLAT Index



Inverted File FLAT

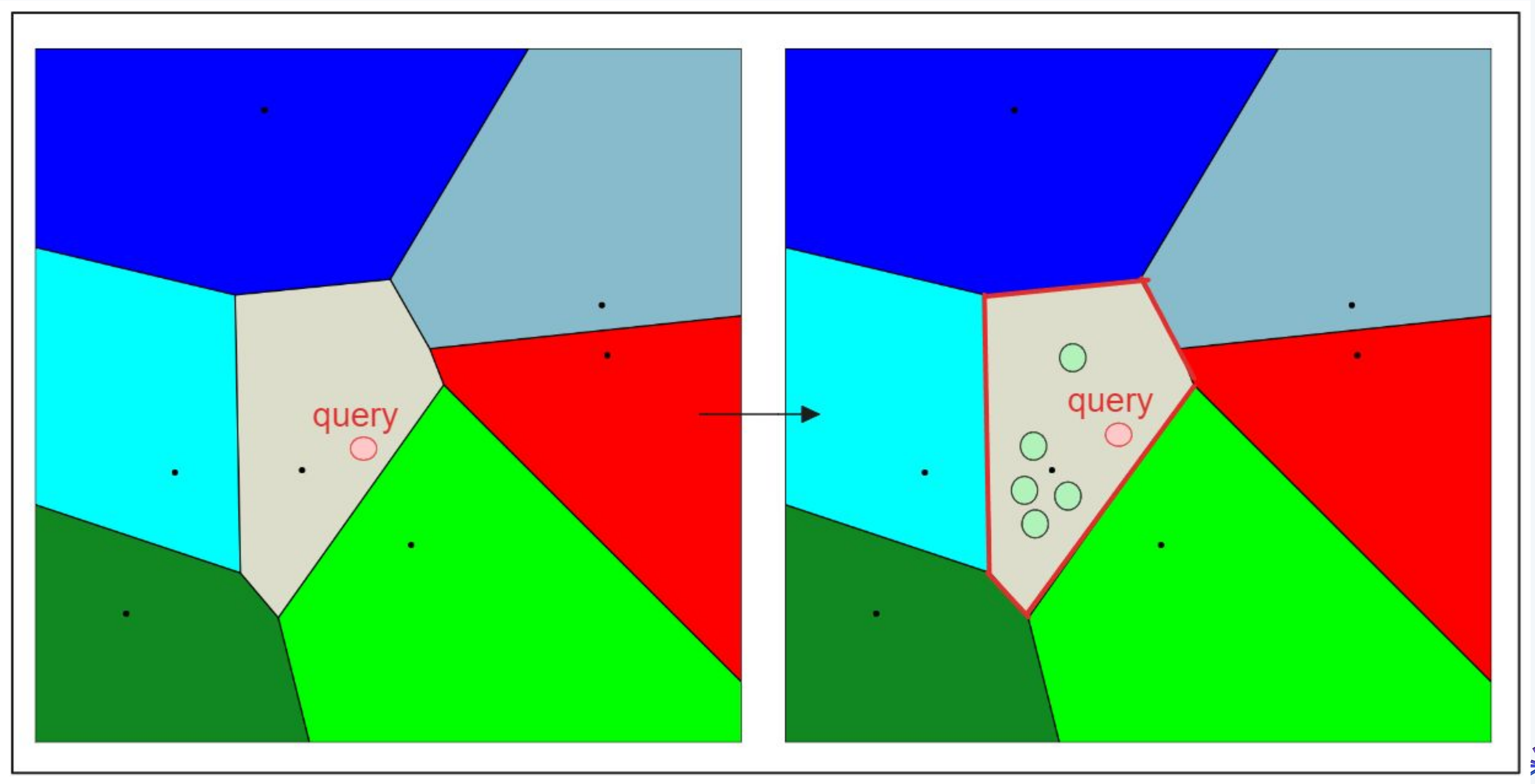
(IVF-FLAT)

IVF-FLAT Index



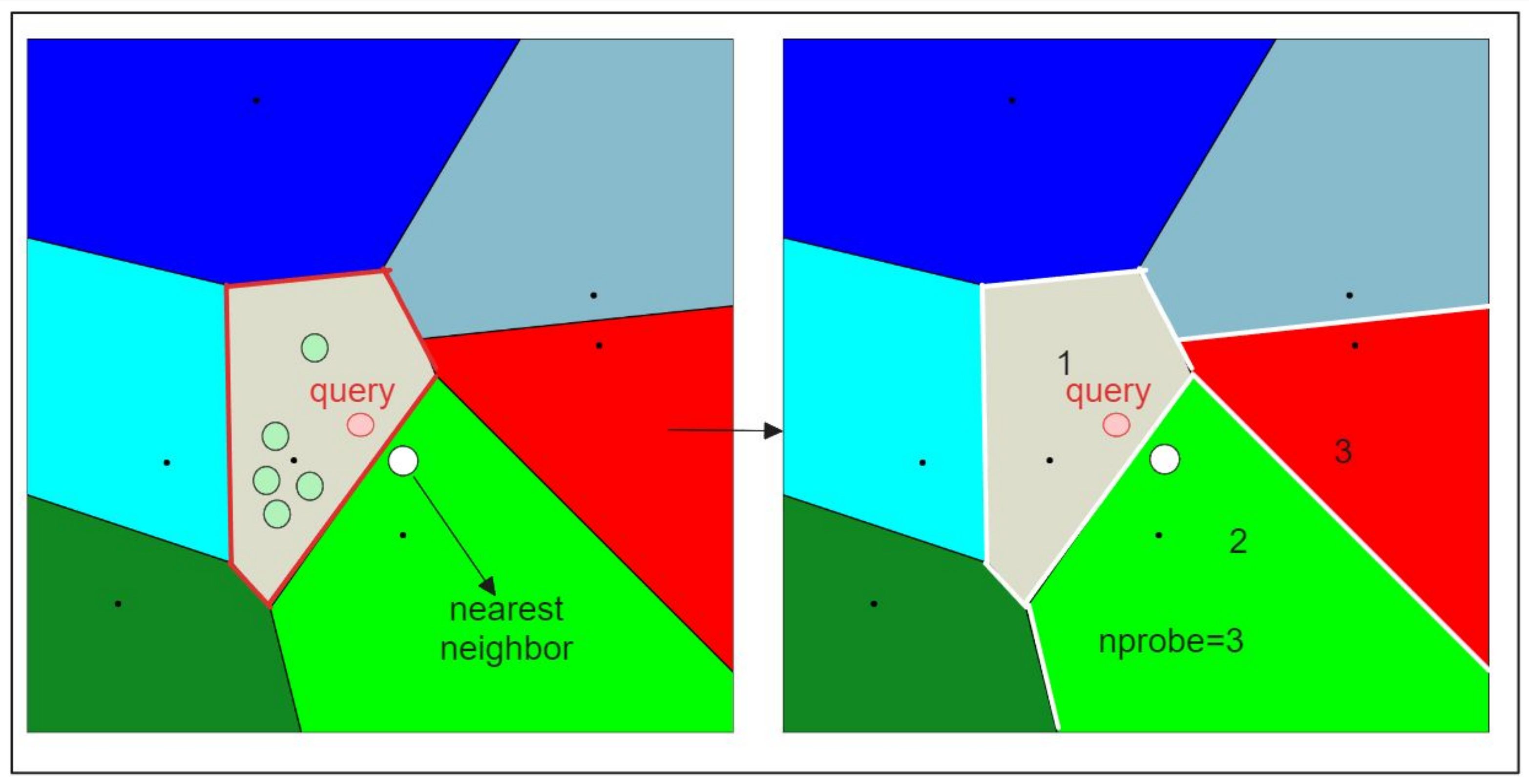
zilliz

IVF-FLAT Index



zilliz

IVF-FLAT Index



zilliz

Inverted File with Quantization

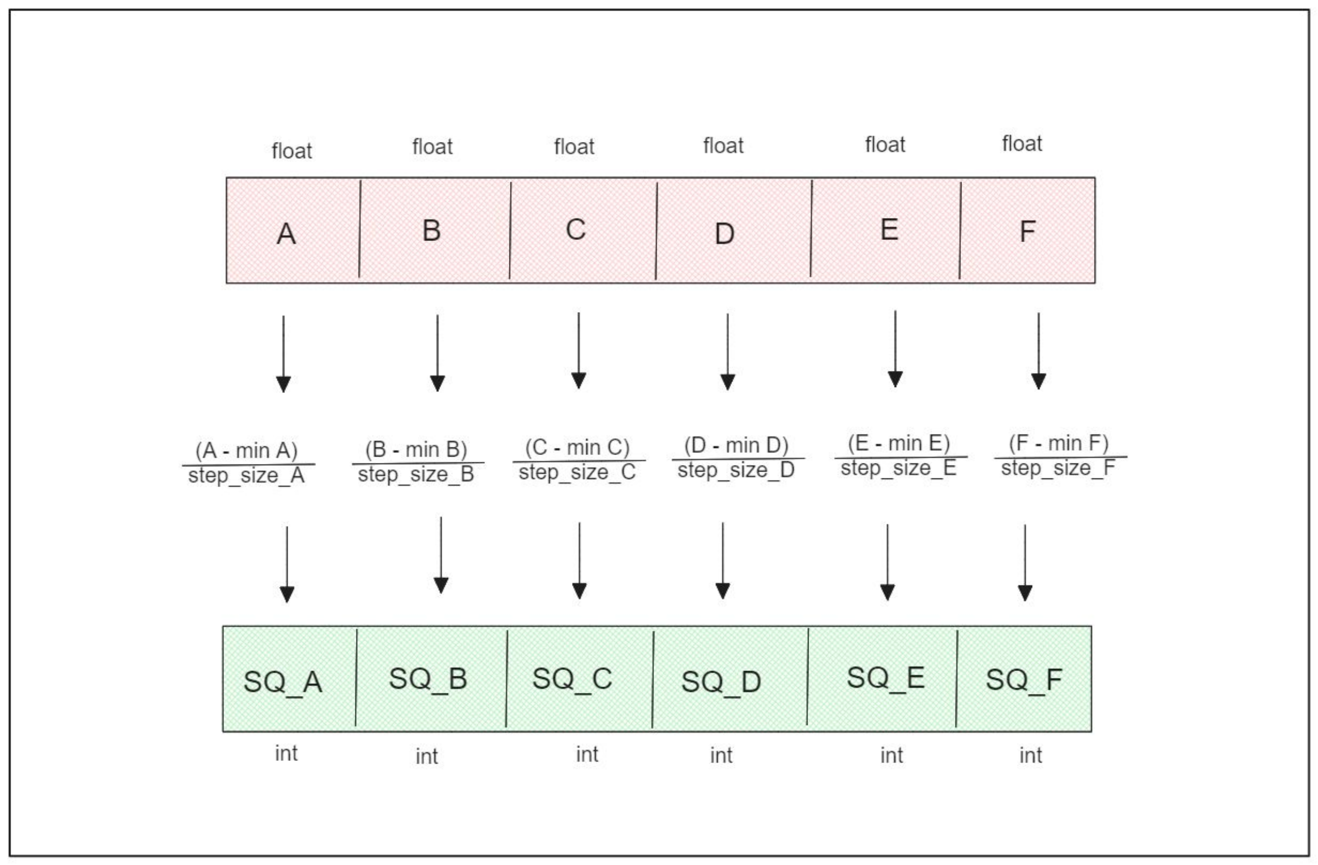
zilliz

Scalar Quantization (IVF-SQ8)

- Mapping Float ⇒ Integers

$$\text{step size} = \frac{\text{max\_value - min\_value}}{\text{number of bins}}$$

Scalar Quantization (IVF-SQ8)

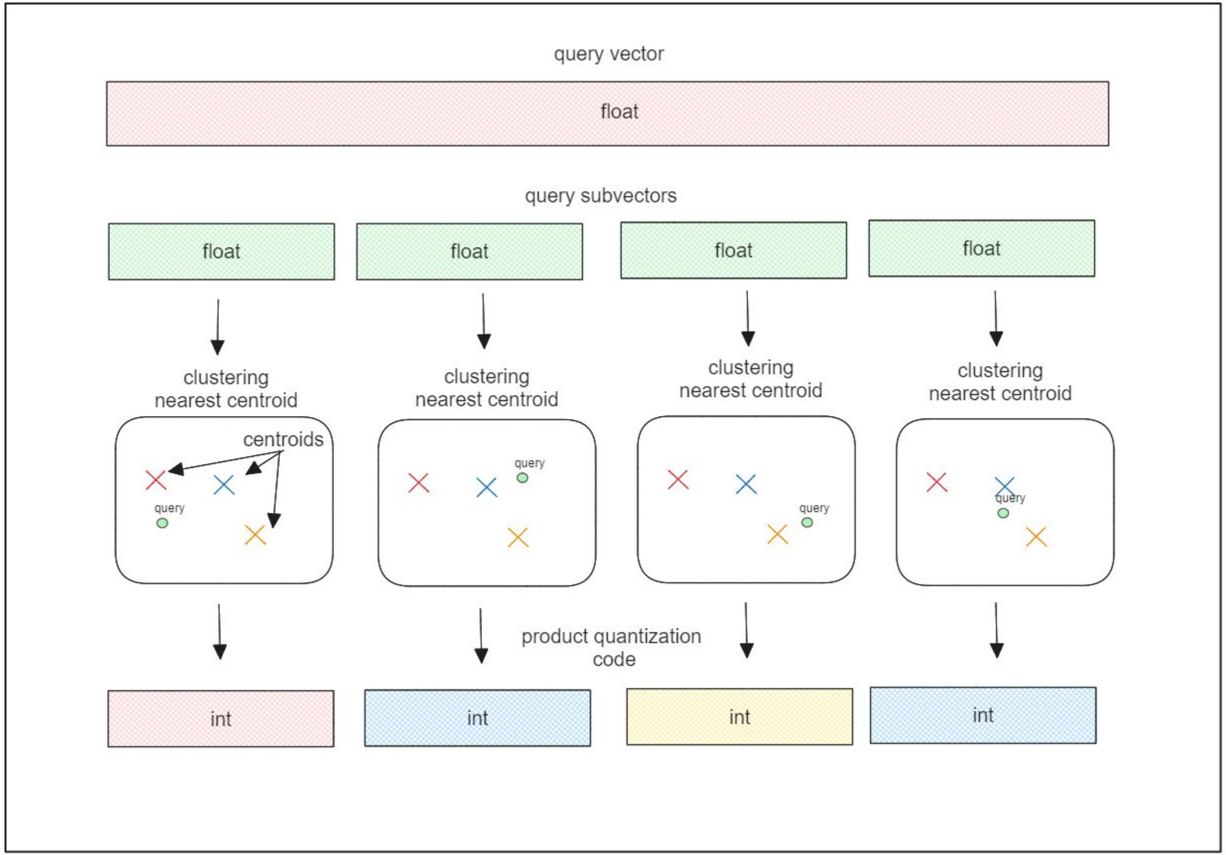


Product Quantization (IVF-PQ)

- Consider the distribution of each dimension
- Divides vector embeddings into subvectors
- Performs clustering within each subvector to create centroids
- Encodes each subvector with the ID of the nearest centroid

 |

Product Quantization (IVF-PQ)

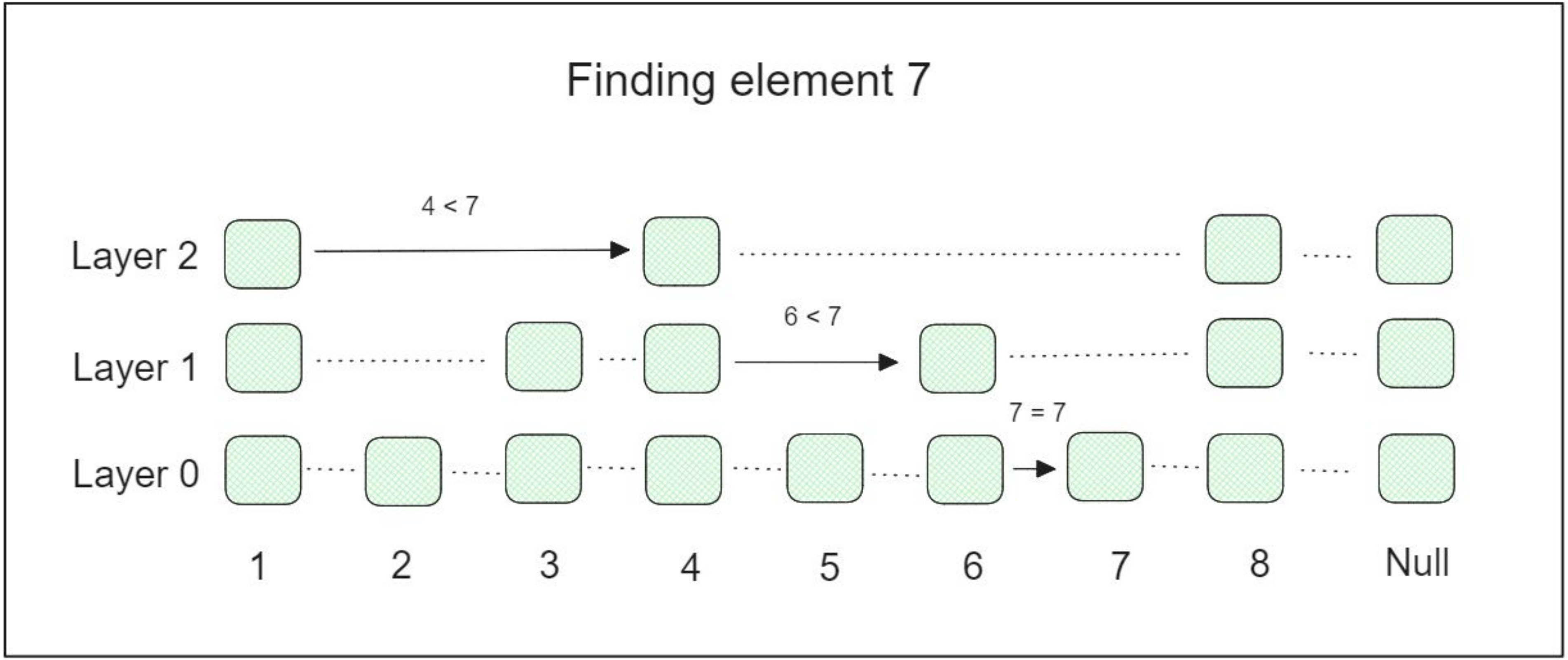


 |

zilliz

Hierarchical Navigable Small World (HNSW)

HNSW - Skip List



HNSW - NSW Graph

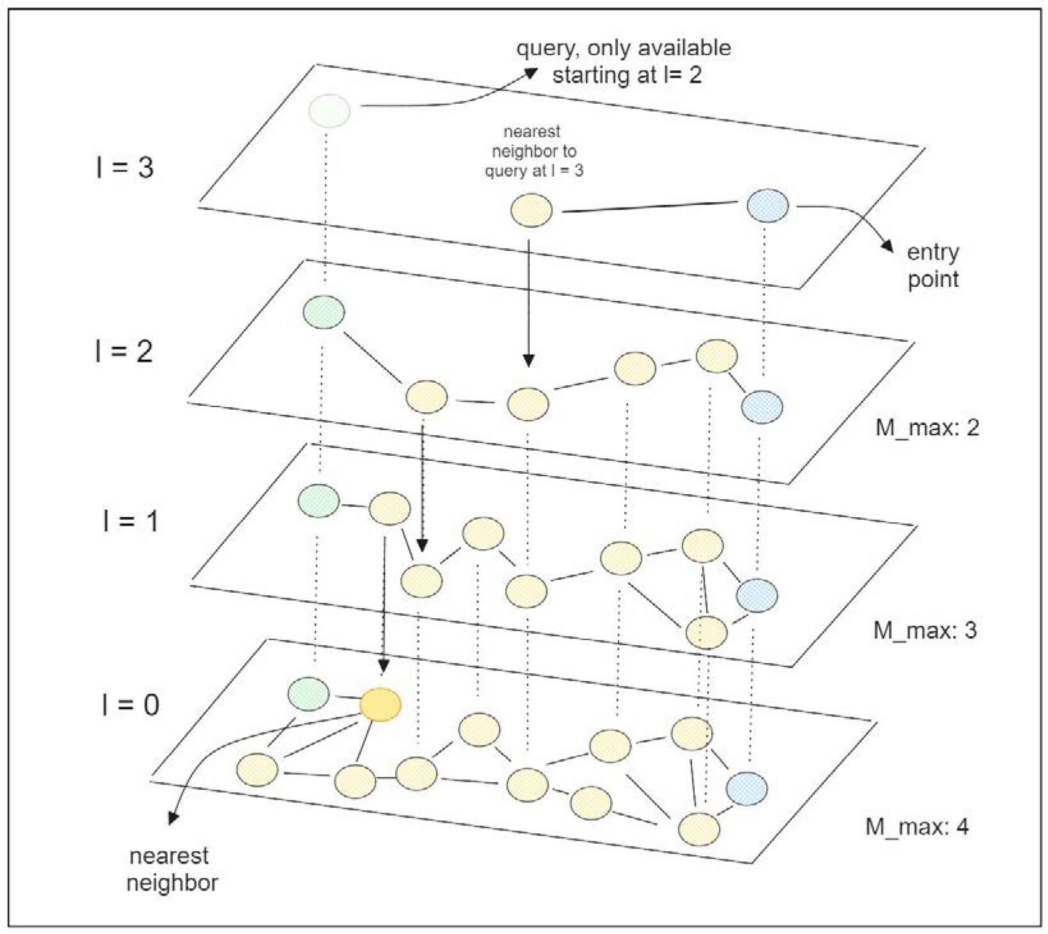
- Built by randomly shuffling data points and inserting them one by one, with each point connected to a predefined number of edges (M).

  ⇒ Creates a graph structure that exhibits the "small world".

  ⇒ Any two points are connected through a relatively short path.

zilliz

HNSW



query, only available starting at l= 2
nearest neighbor to query at l = 3
l = 3
entry point
l = 2
M_max: 2
l = 1
M_max: 3
l = 0
M_max: 4
nearest neighbor

Category	Index	Accuracy	Latency	Throughput	Index Time	Cost
Graph-based	Cagra (GPU)	High	Low	Very High	Fast	Very High
	HNSW	High	Low	High	Slow	High
	DiskANN	High	High	Mid	Very Slow	Low
Quantization-based or cluster-based	ScaNN	Mid	Mid	High	Mid	Mid
	IVF_FLAT	Mid	Mid	Low	Fast	Mid
	IVF + Quantization	Low	Mid	Mid	Mid	Low

 |

zilliz

Picking an Index

- 100% Recall – Use `FLAT` search if you need 100% accuracy
- 10MB < `index_size` < 2GB – Standard IVF
- 2GB < `index_size` < 20GB – Consider `PQ` and `HNSW`
- 20GB < `index_size` < 200GB – Composite Index, `IVF_PQ` or `HNSW_SQ`
- Disk-based indexes

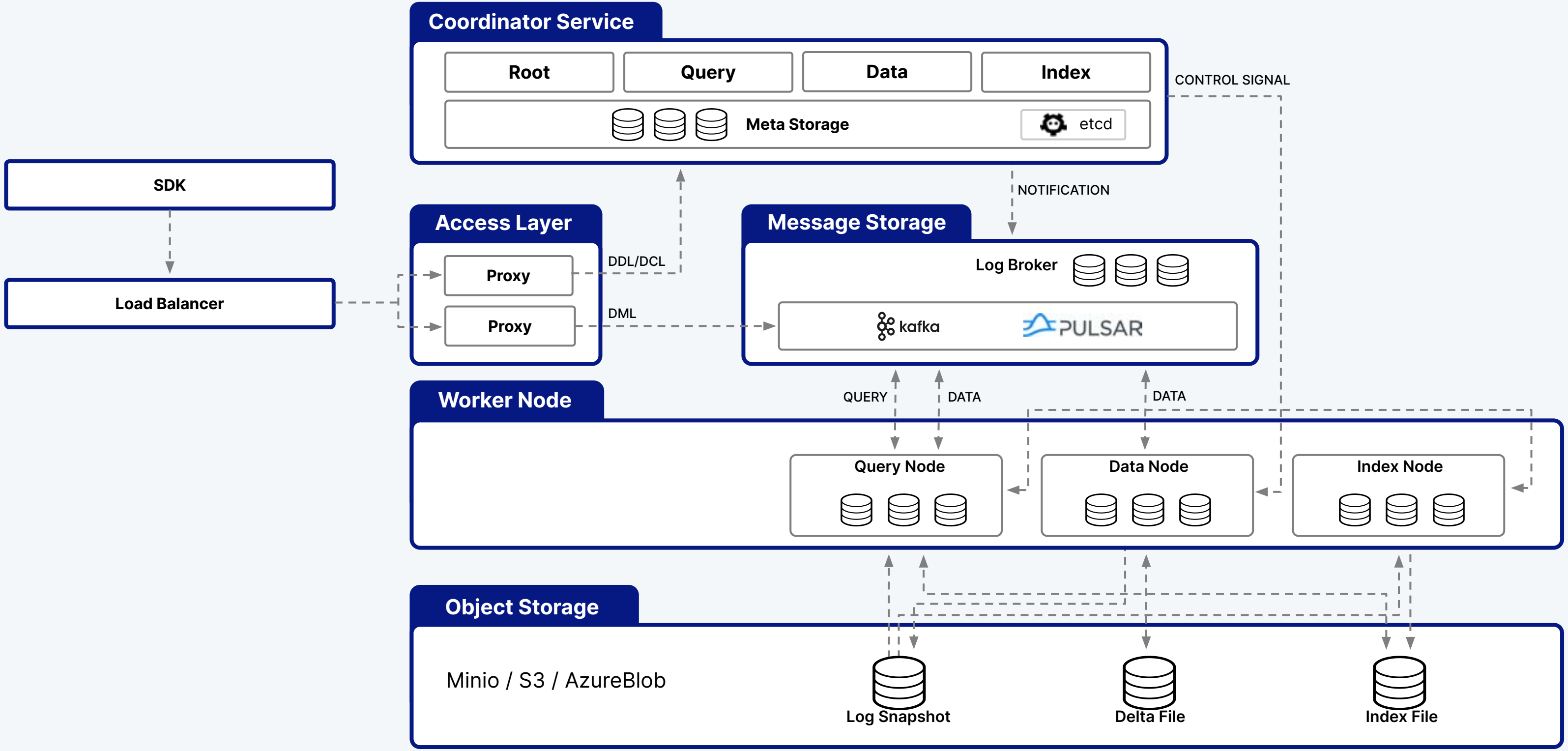
 |

07

Milvus Architecture

zilliz

High-level overview of Milvus' Architecture



 |

Questions?

Check our Github

github.com/milvus-io/

Chat with me on Discord!

discord.gg/FG6hMJStWu

 |