zilliz | NVIDIA

Milvus + RAPIDS RAFT

Accelerating RAG and vector search workflows to improve high efficiency search, data freshness, and recall.

Charles Xie - *CEO, Zilliz*

Corey J. Nolet - *Principal Engineer, Nvidia*

Charles Xie

Founder and CEO of Zilliz, creator of Milvus

- Board member of LF AI & Data Foundation and chairperson from 2020 to 2021.
- Founding engineers of Oracle 12c cloud database.
- Master in Computer Science from University of Wisconsin-Madison.

Corey Nolet

Principal engineer at Nvidia

- Lead engineering for vector search, machine learning, and data mining primitives
- Built massive-scale analytics platforms for the defense industry prior to Nvidia.
- Masters degree in computer science from Johns Hopkins.
- Computer science PhD candidate, University of Maryland, Baltimore County.

AGENDA

01 Vector Database and RAG

02 GPU Powered Milvus

03 Performance Evaluation

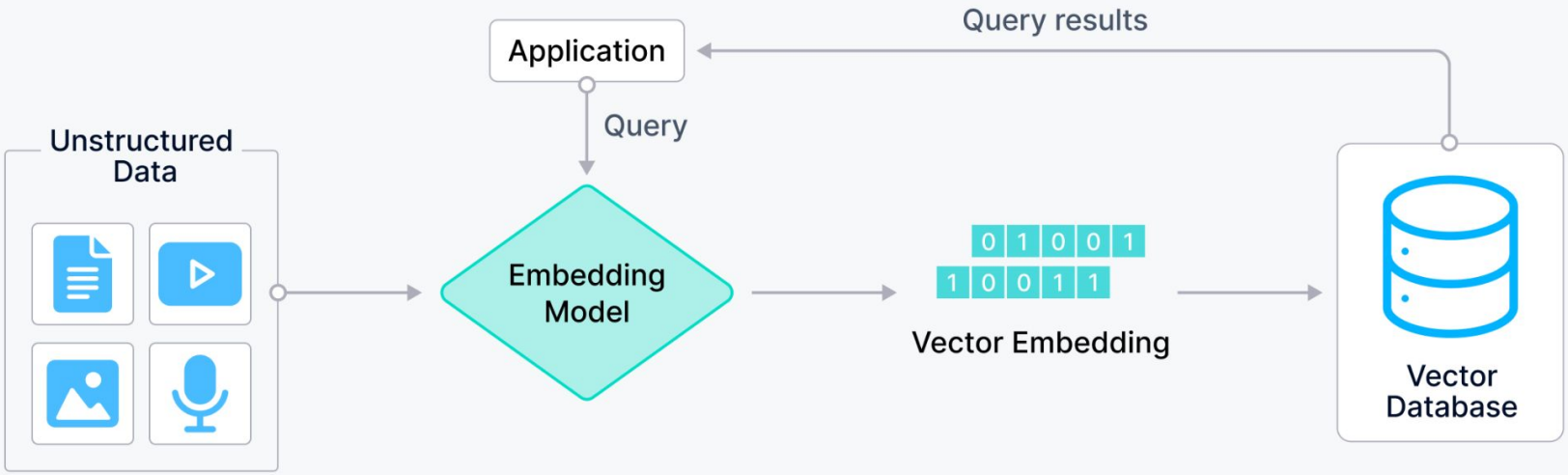
04 Introducing RAPIDS cuVS

01

Vector Database and RAG

What is a Vector Database?

A vector database stores embedding vectors and enables semantic search across different types of unstructured data.



Milvus: the world's most widely-adopted vector database

26K+
GitHub Stars

260+
Contributors

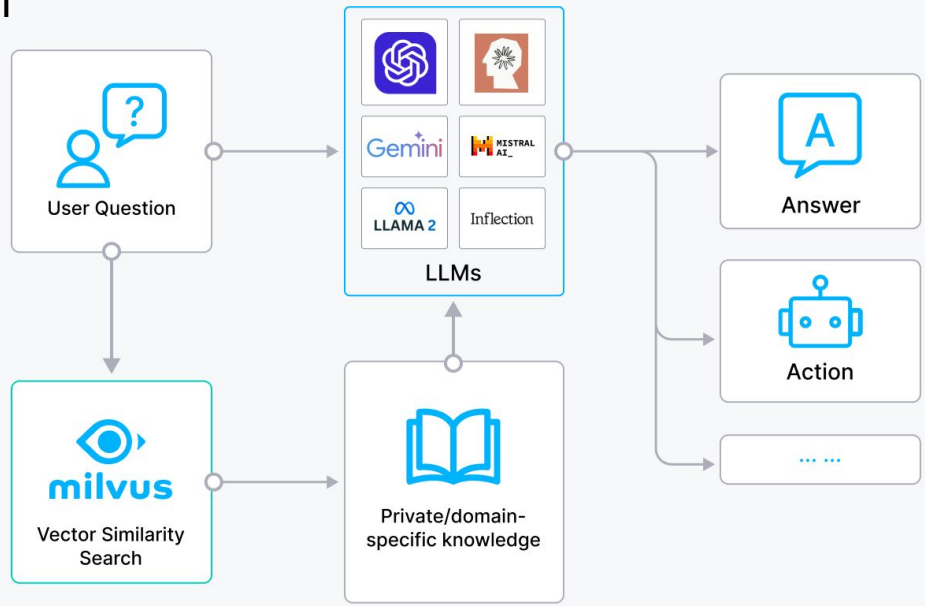
5K+
Enterprise users

20M+
Downloads

Retrieval-Augmented Generation (RAG)

A technique that combines the strength of retrieval-based and generative models:

- Improve accuracy and relevance
- Provide private/domain-specific knowledge
- Eliminate hallucination



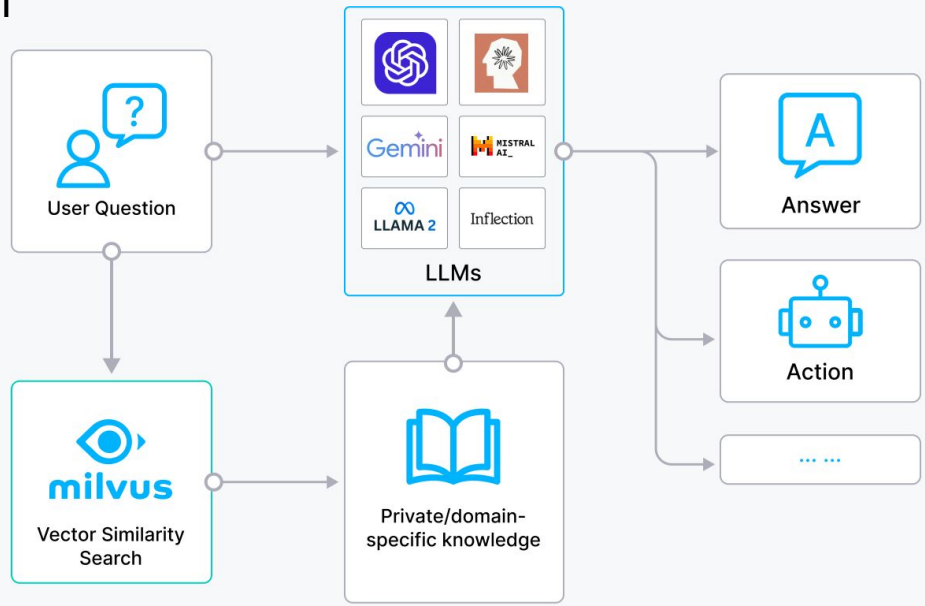
Retrieval-Augmented Generation (RAG)

A technique that combines the strength of retrieval-based and generative models:

- Improve accuracy and relevance
- Provide private/domain-specific knowledge
- Eliminate hallucination

Challenges:

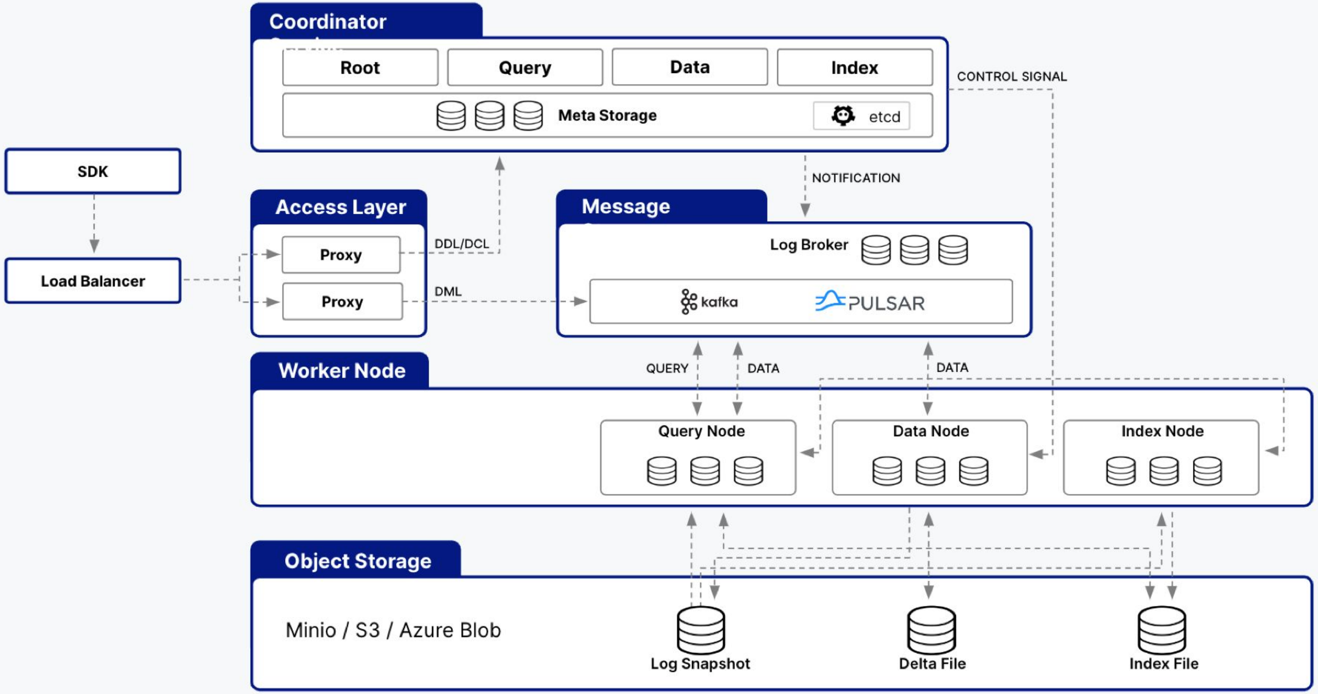
- Data refreshness
- Throughput



02

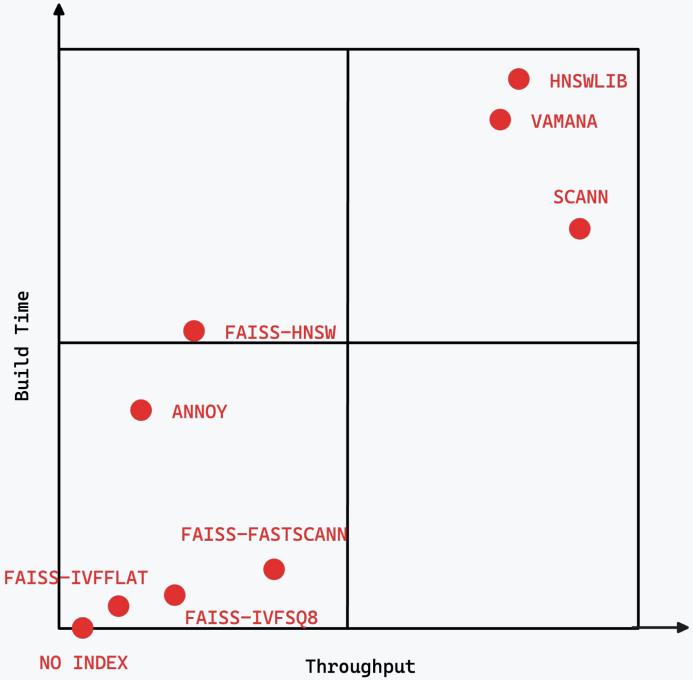
GPU Powered Milvus

Milvus 2.0 Architecture



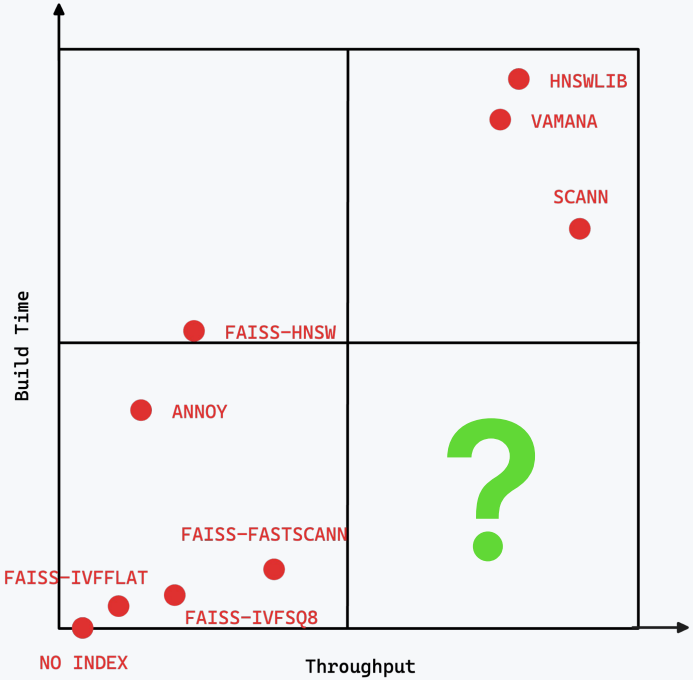
Milvus without GPU acceleration

- Indexes with high throughput often require more time to construct

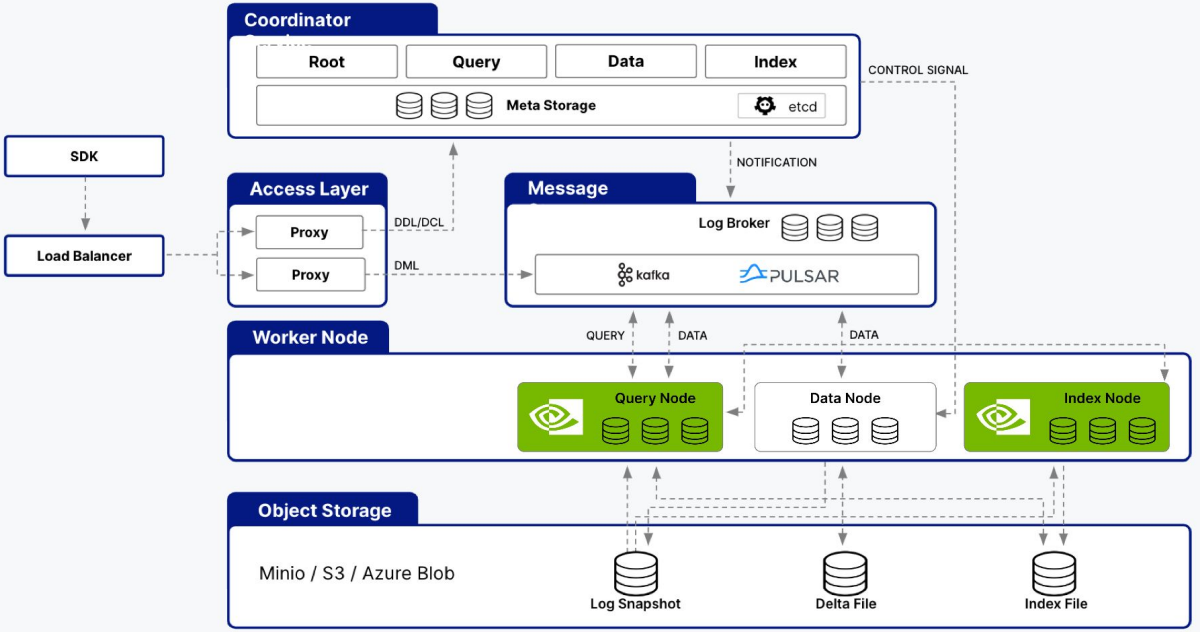


Milvus without GPU acceleration

- Indexes with high throughput often require more time to construct

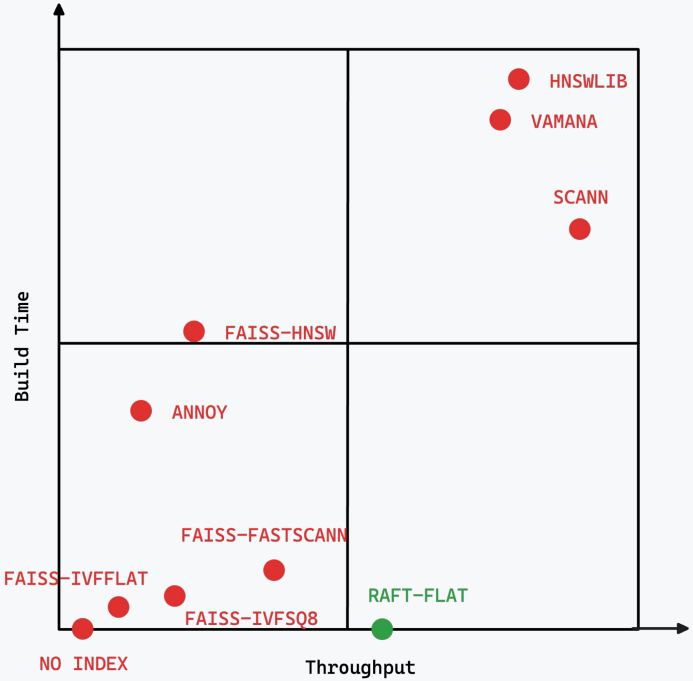


GPU Powered Milvus



Milvus with GPU acceleration

High throughput and rapid construction



zilliz NVIDIA

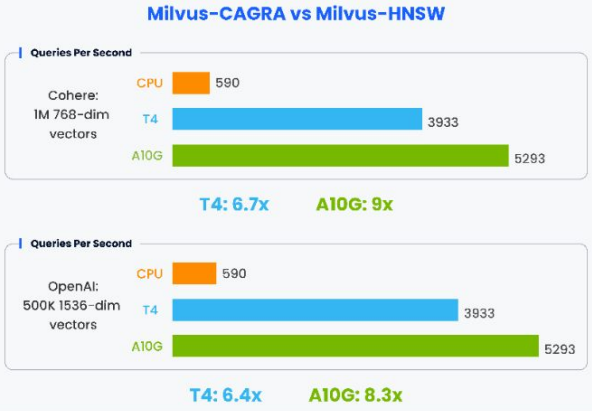
Miluvs with CAGRA

Super high throughput and rapid construction

03

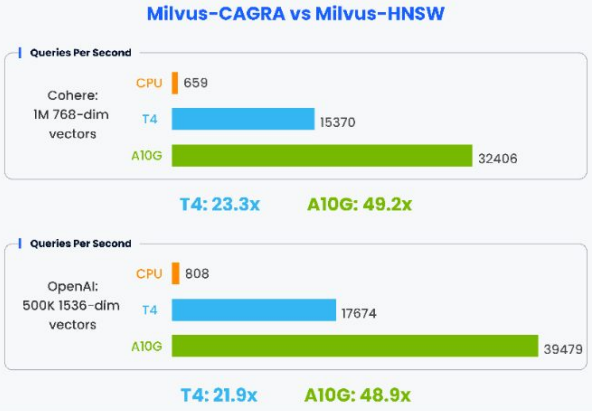
Performance Evaluation

Similarity Search



Batch Size = 1

Batch Size = 10

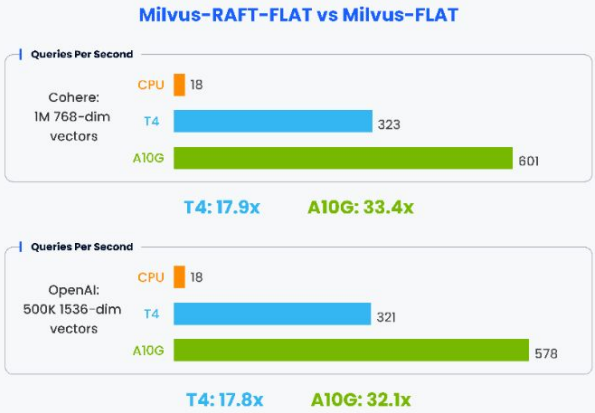


Batch Size = 100

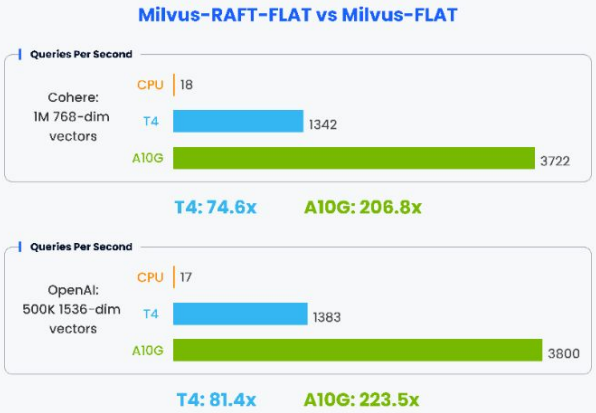
CPU: m6id.2xlarge T4: g4dn.2xlarge A10G: g5.2xlarge Top 100 Recall: 98%
Dataset:: https://github.com/zilliztech/VectorDBBench

zilliz NVIDIA

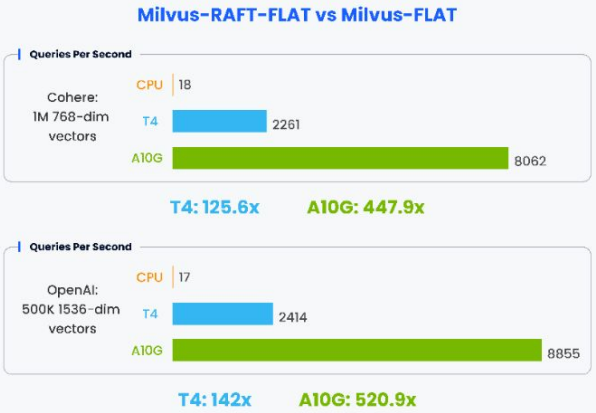
Exact Search



Batch Size = 1



Batch Size = 10



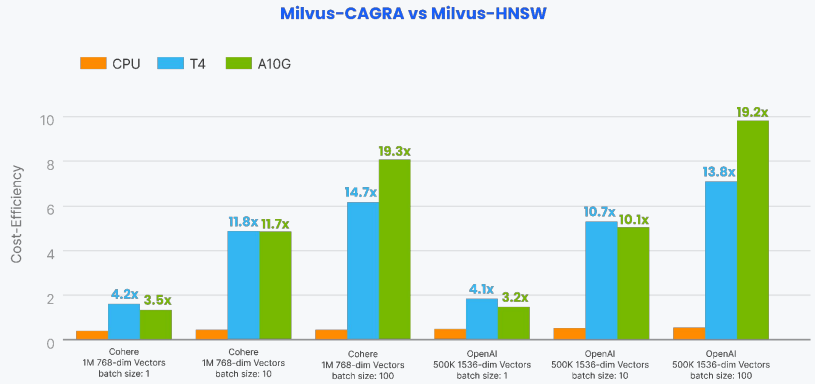
Batch Size = 100

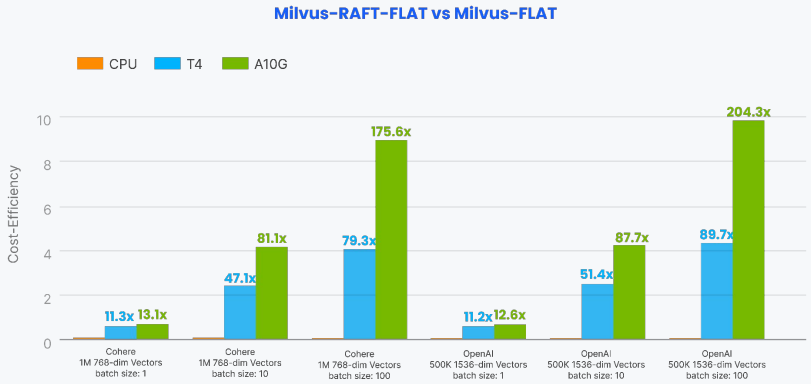
CPU: m6id.2xlarge T4: g4dn.2xlarge A10G: g5.2xlarge Top 100 Recall: 98%
Dataset:: https://github.com/zilliztech/VectorDBBench

zilliz NVIDIA

Cost Efficiency

	Instance type	Price($/h)
T4	g4dn.2xlarge	0.752
A10G	g5.2xlarge	1.212
CPU	m6id.2xlarge	0.4746





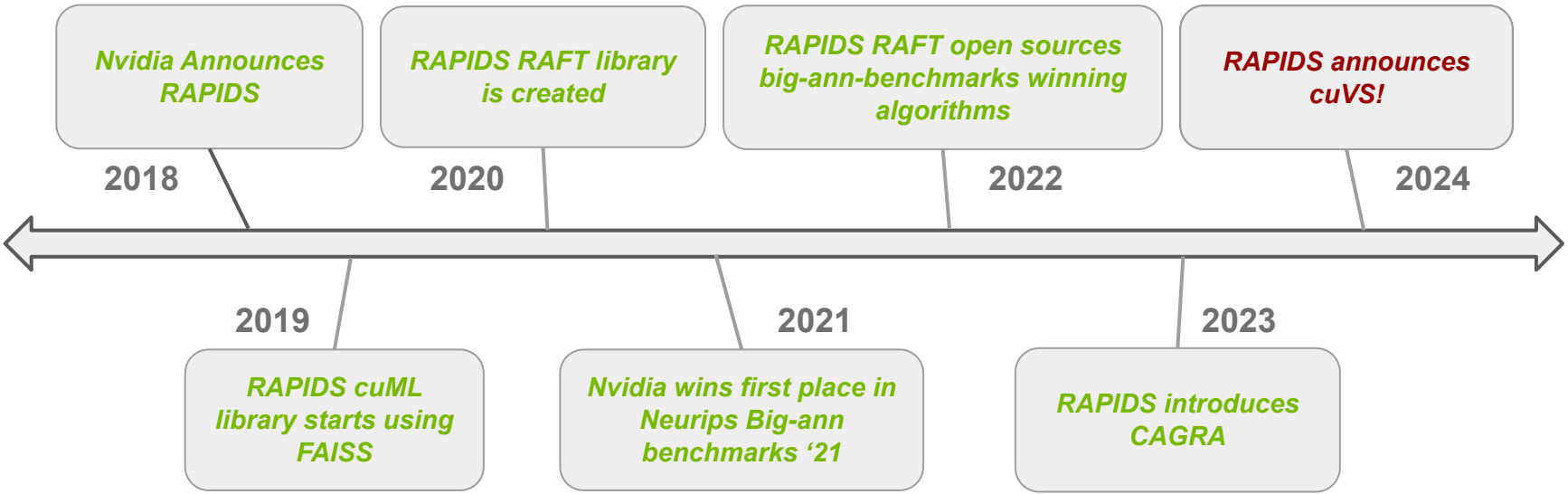
zilliz NVIDIA

04

Introducing RAPIDS cuVS

Vector Search Timeline

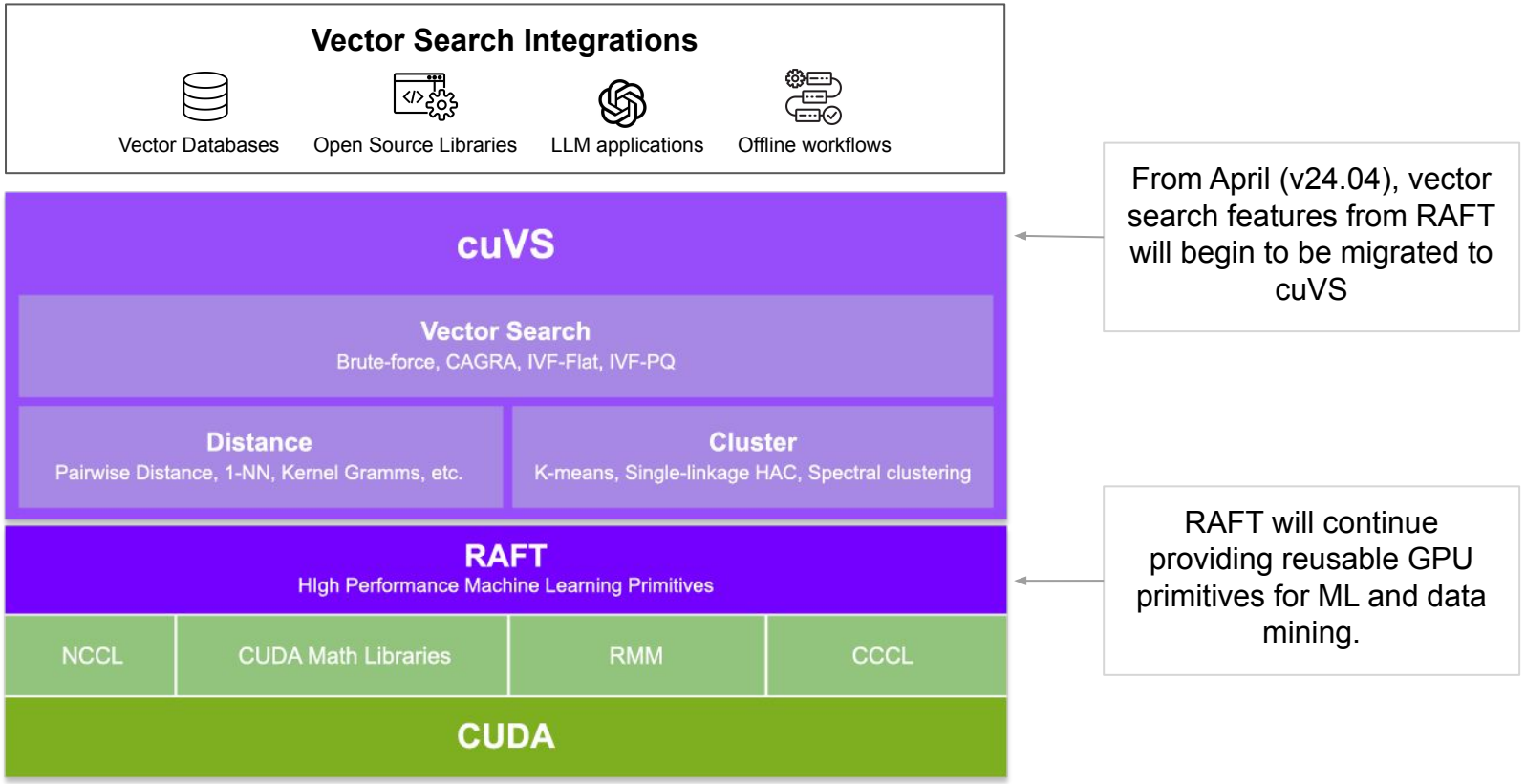
Brief history of GPU-accelerated nearest neighbors at Nvidia



zilliz NVIDIA

RAPIDS cuVS Overview

GPU-Accelerated Vector Search



From April (v24.04), vector search features from RAFT will begin to be migrated to cuVS

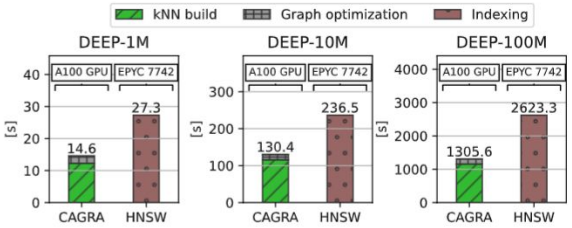
RAFT will continue providing reusable GPU primitives for ML and data mining.

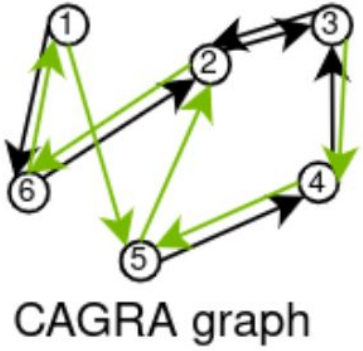
zilliz NVIDIA

CAGRA

GPU-Accelerated State-of-the-Art Graph-Based ANN

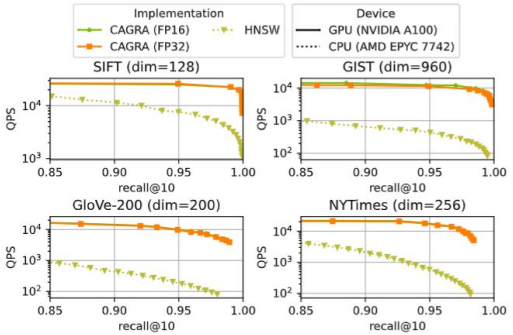
- ***Individual queries*** parallelized during search
- Setting records for both ***single query*** and ***large batch*** performance
- ***Higher throughput*** than existing GPU Graph ANNs and ***lower latency*** than SOTA CPU Graph ANNs



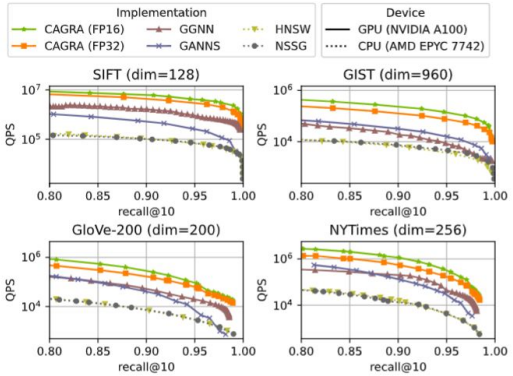


CAGRA graph

Single query at a time



Batches of 10k queries



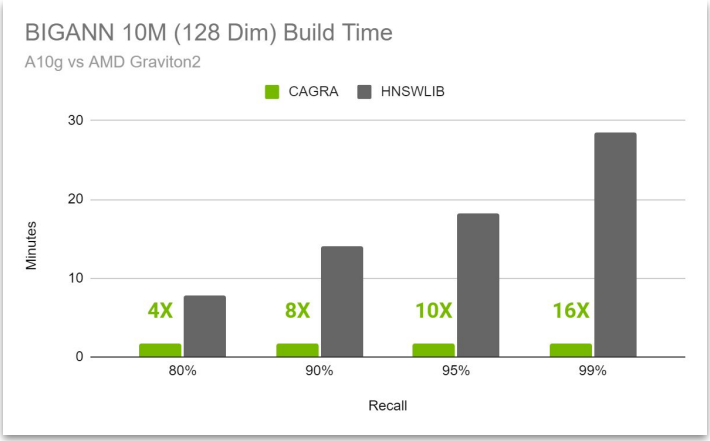
"CAGRA: Highly Parallel Graph Construction and Approximate Nearest Neighbor Search on the GPU", Ootomo et al., 2023

CAGRA Index Build Times

Compared to HNSW

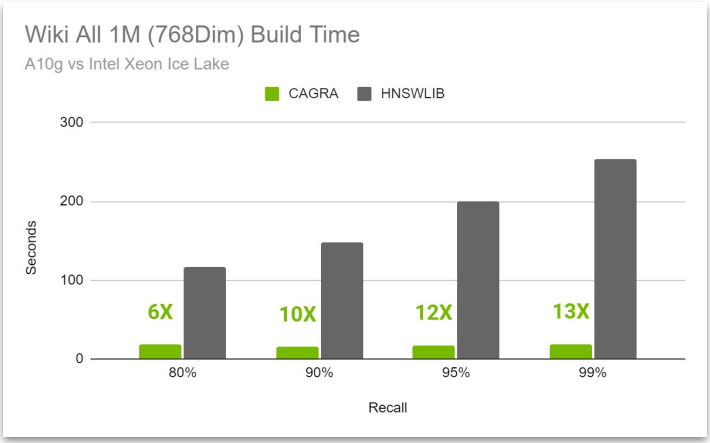
Value of using GPU for index build increases with

- Recall
- Dimensionality
- Number of vectors



CPU	GPU
AMD EPYC	A10g 24GB
$0.81/hr	**$1.21/hr**
16 vCPU	8 vCPU
128GB Memory	32GB Memory

GPU($) / CPU($) = 1.50
Tested on AWS



CPU	GPU
Intel Ice Lake	A10g 24GB
$1.54/hr	**$1.21/hr**
32 vCPU	8 vCPU
128GB Memory	32GB Memory

GPU($) / CPU($) = 0.786
Tested on AWS

zilliz NVIDIA

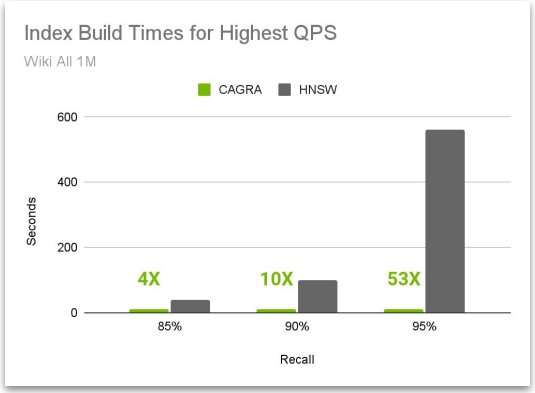
CAGRA+HNSW

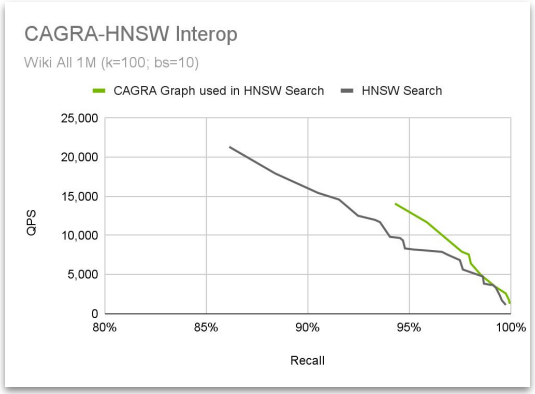
Building index on GPU and searching on CPU

Index build can be offloaded to GPU and search performed on CPU using HNSW

- Building HNSW indexes is slow – can take hours or days to build
- CAGRA indexes can be built 20x faster than HNSW
- HNSW can search a graph built with CAGRA
- CAGRA graph can outperform HNSW on CPU search, especially at larger dimensions

This feature will soon be available in Milvus!

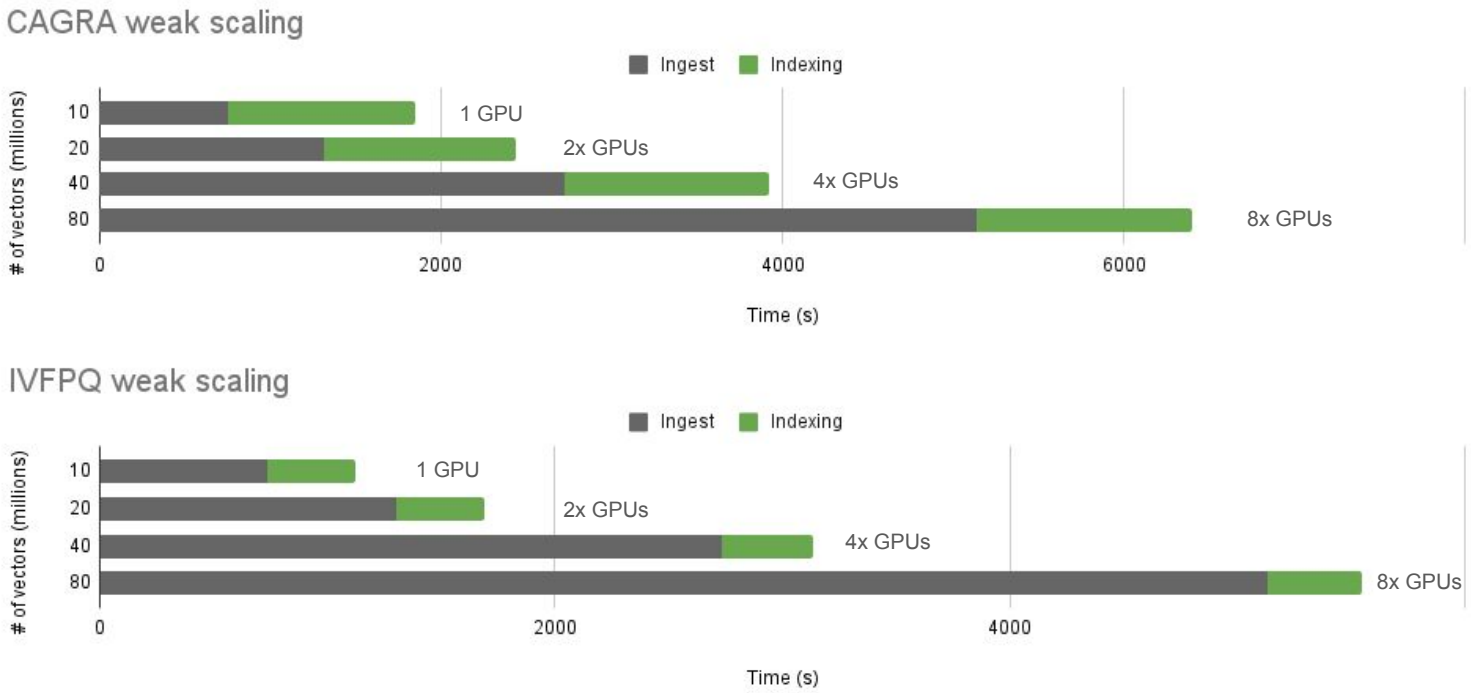




zilliz NVIDIA

Measuring ingest at scale

Scaling Milvus up on a DGX H100 with Multiple GPUs (1024 dimensions)



Ingest includes writing vectors to local object storage and segment files

zilliz NVIDIA

Measuring ingest at scale

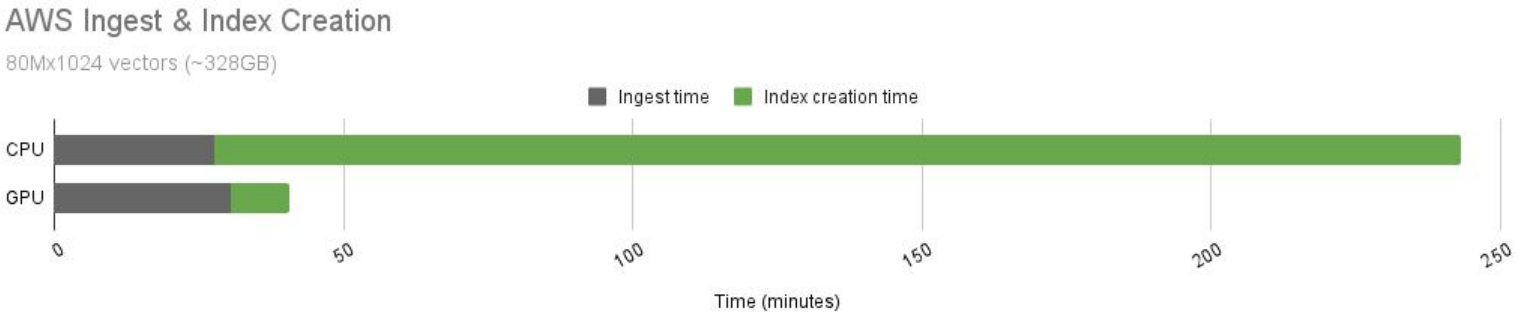
Scaling Milvus up on a DGX H100 with Multiple GPUs (1024 dimensions)

Ingest includes writing vectors to local object storage and segment files

zilliz NVIDIA

Measuring ingest at scale

Cloud cost performance 80M x 1024 vectors (~328GB)



CPU r6id.32xlarge	GPU g5.48large
$9.68/hr	$16.29/hr
Intel Ice Lake 8375C	AMD EPYC
	8x A10G (192GB total)

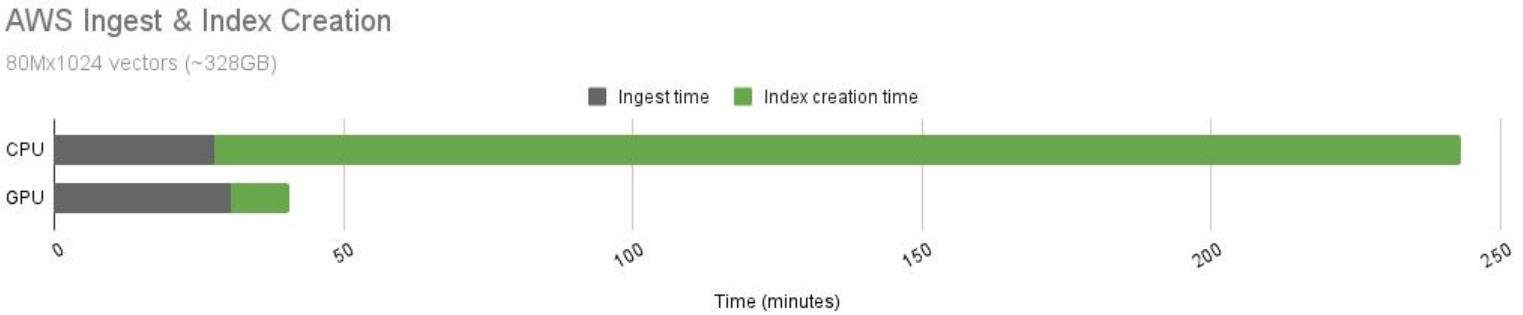
GPU($) / CPU($) = 1.68
Tested on AWS

Build indexes ~21x faster @ 68% higher cost per hour

zilliz NVIDIA

Measuring ingest at scale

Cloud cost performance 80M x 1024 vectors (~328GB)



CPU r6id.32xlarge	GPU g5.48large
$9.68/hr	$16.29/hr
Intel Ice Lake 8375C	AMD EPYC
	8x A10G (192GB)

GPU($) / CPU($) = 1.68
Tested on AWS

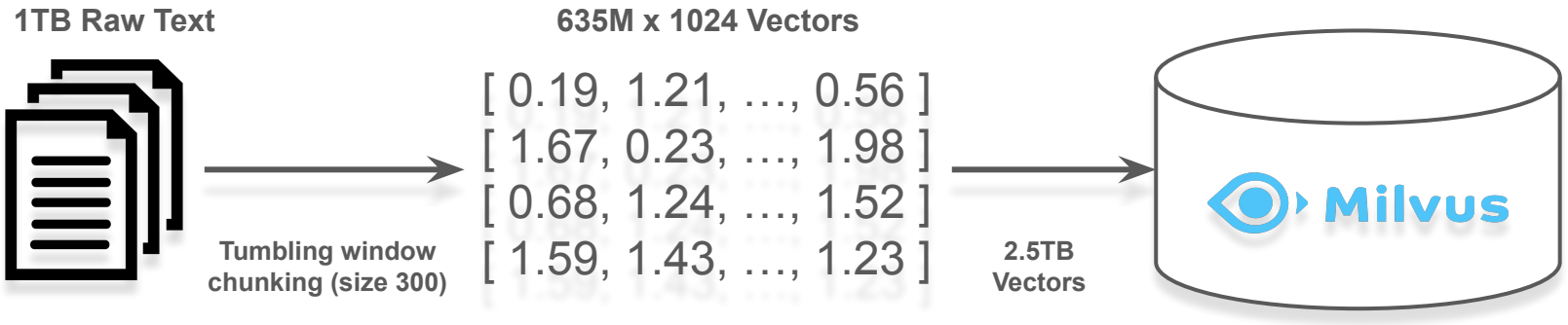
Build indexes ~21x faster @ 68% higher cost per hour

Build 12.5x faster @ the same cost per hour

zilliz NVIDIA

Measuring ingest at 1TB scale

Massive-scale indexing on a DGX H100 (8x GPUs)



Ingest time: 11.5 hours

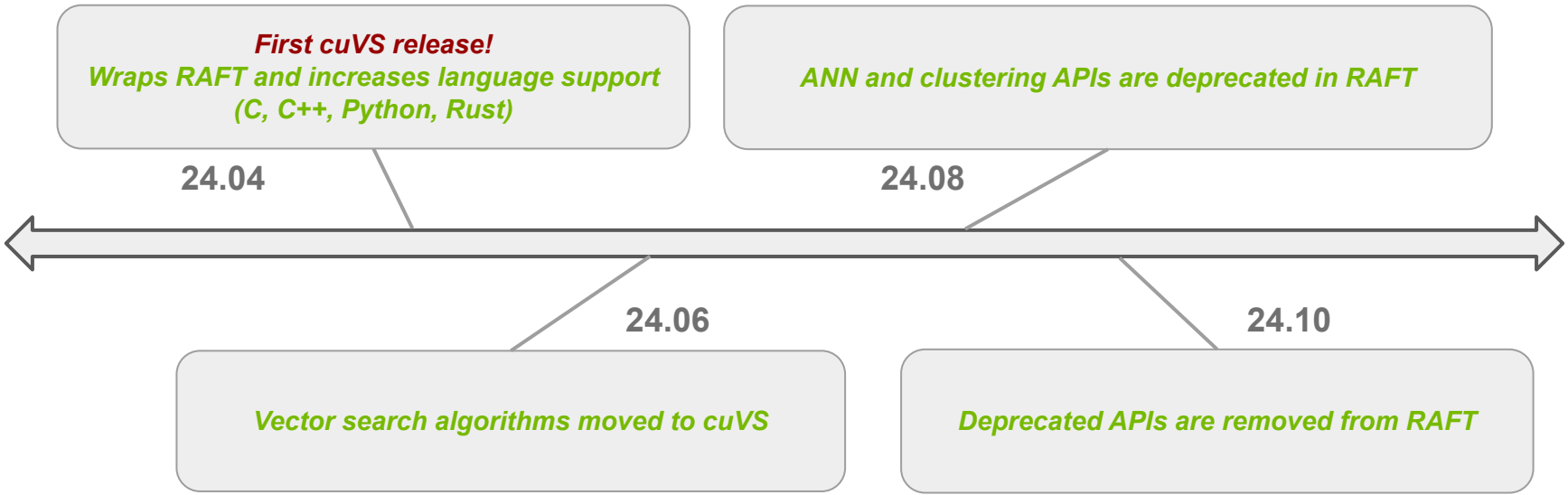
Index creation time (GPU): 56 minutes

Index creation time (CPU estimated): 6¼ days (20x speedup)

zilliz NVIDIA

Migration timeline

Moving vector search algorithms from RAFT to cuVS



Getting Started with RAPIDS cuVS

A library for vector search and clustering on the GPU

cuVS https://rapids.ai/cuvs

FAISS https://github.com/facebookresearch/faiss/blob/HEAD/INSTALL.md

Milvus https://milvus.io/docs/install_standalone-docker-compose-gpu.md

Kinetica https://www.kinetica.com/try/#download_instructions

Lucene https://github.com/SearchScale/lucene-cuvs

Ecosystem Partners

SearchScale · Milvus · FAISS · kinetica · Solr · Weaviate · redis · APACHE LUCENE · DATASTAX · OpenSearch

Thank You

https://zilliz.com

https://github.com/milvus-io/milvus

https://milvus.io/discord

https://rapids.ai/cuvs

https://github.com/rapidsai/cuvs

https://rapids.ai/slack-invite/