zilliz

Why Vector Databases Matter for Unstructured Data

TECHNICAL PAPER

Release Date

October, 2024

Table of Contents

Introduction

As the amount of data exponentially grows in the information age, unstructured data is exploding. Images, videos, texts, medical data, and housing data are different kinds of unstructured data that are experiencing massive growth today. In addition, smartphones, IoT devices, and social media contribute to the rapid rise of unstructured data. IDC predicts that 90% of data will be unstructured by 2025, and per IDC's Global DataSphere 2022, unstructured data growth from 2021 to 2022 is forecast to grow more than 9x the volume of structured data. Machine learning techniques can transform unstructured data into feature vectors. This technique makes it possible to analyze and manage the unstructured data so many current technologies produce and rely on.

These vectors are often huge and can vary from tens to hundreds of dimensions. A vector database needs to be able to handle vectors of that size and to be flexible. In addition, the amount of vector data in the world is growing as the amount of unstructured data grows, and the need for a scalable, dynamic vector database is high. This paper will explain unstructured data, popular use cases for unstructured data, and how Milvus differs from other vector data management systems. The founders of Milvus have published an academic paper that goes into more of the technical details behind Milvus, which can be found here.

What is Unstructured Data?

Unstructured data refers to information that lacks a predefined structure or data model. Unlike structured data, which is neatly organized into rows and columns (like in traditional databases or spreadsheets), unstructured data is far more freeform and doesn’t fit into conventional formats. This category of data includes text (emails, documents, social media posts), images, videos, audio files, and even more complex forms of information such as web pages and sensor data.

While unstructured data can offer valuable insights for businesses, extracting that value has been a persistent challenge. A 2019 Deloitte survey found that only 18% of organizations reported successfully leveraging unstructured data. This isn’t due to a shortage of such data; unstructured data is abundant. The problem lies in the lack of tools and technologies capable of processing and analyzing this diverse and disorganized resource. The sheer volume and irregular nature of unstructured data make it difficult for traditional programs and databases to handle, discouraging many organizations from even attempting to tap into the potential insights hidden within it.

> *According to a 2019 survey by Deloitte, only 18% of organizations reported being able to take advantage of unstructured data.*

Fortunately, advances in AI and machine learning (ML) have changed the landscape, offering new ways to extract meaning from unstructured data. For example, ML models can convert unstructured data into feature vectors—a numerical representation of objects. Feature vectors represent key characteristics of an object, such as the color or edges of an image, the structure of text, or the volume of sound. Once data is represented as vectors, mathematical functions can be applied to analyze it. One common approach is to measure the distance between two vectors to compare their similarity.

These vectors often have extremely high dimensions and require specialized databases to store and process them. Traditional databases struggle to handle the size and dynamic nature of these vectors, as they aren't designed for the frequent reads and writes necessary for working with unstructured data at scale. As a result, organizations increasingly rely on vector databases purpose-built for managing the complexities of unstructured data.

Popular Use Cases for Unstructured Data

Large Language Model (LLM) Training

The rise of large language models (LLMs), such as OpenAI’s GPT and Google’s Gemini series, has redefined almost every industry by significantly enhancing how we process and interact with information. These models are trained primarily on vast amounts of unstructured text data, enabling them to understand and generate natural language responses. During pre-training, LLMs learn from massive datasets that include raw text from sources like web pages, books, and articles. This extensive exposure allows them to generalize across diverse linguistic inputs, making them powerful tools for tasks like summarization, translation, and content generation.

However, not all AI models are limited to text alone. Multimodal LLMs, such as OpenAI's CLIP and Google's Gemini, extend these capabilities by incorporating a wide range of unstructured data from multiple sources, including images, audio, and videos. These models can process diverse inputs, enabling them to generate text-based descriptions for images, understand video content, or even create audio captions. This broader scope makes multimodal LLMs highly versatile, allowing them to handle more complex applications such as video summarization or cross-media search. Their ability to work across different data formats makes them a critical use case for effectively leveraging unstructured data across various contexts.

Retrieval-Augmented Generation (RAG)

While LLMs and multimodal AI models are powerful, their knowledge is confined to what they were pre-trained on, which can be outdated or lack specific real-time information. Retrieval Augmented Generation (RAG) addresses this limitation by connecting these models to external unstructured data sources—such as documents, images, or videos—stored as vector embeddings in vector databases like Milvus and Zilliz Cloud.

When a user submits a query, RAG performs similarity searches on this external unstructured data, retrieving relevant information that the model can use as context to generate a more accurate response. This makes RAG an indispensable technique for LLM-related tasks requiring up-to-date or domain-specific knowledge from various unstructured data formats.

Natural Language Processing — Question And Answer Chatbots

Chatbots incorporating Natural Language Processing (NLP) have become increasingly popular, as they can effectively imitate a live operator. These chatbots can be programmed to answer users' questions, provide relevant information, and ultimately reduce the need for human labor. One advantage of NLP-powered chatbots is that they can understand and interpret human language, leading to more accurate and personalized responses. Additionally, they can be available 24/7, making them an excellent option for businesses and organizations that want to provide continuous customer support.

A vector database is helpful for a chatbot because it can help interpret user messages and provide relevant responses. Vector databases store words or phrases in a high-dimensional space where each dimension corresponds to a particular feature. By representing words as vectors, chatbots can analyze the relationships between different words and understand the context in which the user asks the question. Representing words as vectors and storing them in a scalable and performant vector database can help the chatbot generate more accurate responses and improve the user experience.

Semantic Search

Semantic search is a way to find results based on the meaning behind a query rather than simply the inputted literal text. Search engines, for example, want to understand a user's intent and context to provide the best search results. So, instead of just searching the internet via keyword matching, search engines now include information like the relationship between the words in a search query, a user's search history and location, and more. Semantic search is how search engines attempt to understand language more humanly to get better results with more context. Contexts like overall global search history, spelling differences, and user information allow search engines to interpret the intent behind a search query and to provide the most relevant results. Search engine companies aren't the only organizations using semantic search; any organization that returns results to user queries, such as online shopping websites, might use semantic search to return relevant results based on the meaning of queries.

Semantic search is handy, but it also adds a lot of data processing into queries that need to return results very quickly. Developers can use machine learning techniques to speed up the semantic search. These techniques transform the text of search queries and context information into vectors. Organizations can then search through vectors to find the best results. You need a well-designed vector data management system to search through vectors quickly and efficiently.

Product Recommendations

Product recommendation is another crucial function for many companies that rely on unstructured data. For example, a company needs to know a user's search and purchase history, what other users purchased and why, and more to make successful recommendations. This information is unstructured data that organizations need to ask complex questions of. Machine learning techniques can turn this data into feature vectors and return accurate and fast results if organizations have a good way of managing the vectors involved.

Anomaly Detection

Anomaly detection is one of the most common goals when working with data. If something unusual is occurring, analysts want to know about it as soon as possible so they can figure out why. For example, you can set up an alert with structured data if a metric rises above a set threshold. However, with unstructured data, vectorization is extremely important, so analysts can use mathematical functions to find anomalies. For example, if a set of images is supposed to look a certain way, it would be very tedious and prone to error for a person to check each one individually manually. However, once each image has been turned into a feature vector, an analyst can write a program to mathematically check for unusual cases.

Managing Unstructured Data with Vector Databases

Unstructured data is inherently complex, and its vector representations are equally vast, capturing that complexity in high-dimensional spaces. When dealing with unstructured data, it's common to encounter billions of vectors, making it essential to have a database that can handle such scale. As a result, the need for scalable vector data management systems has become increasingly critical. These systems must support fast query results on massive datasets and efficiently manage dynamic data, including frequent insertions and deletions.

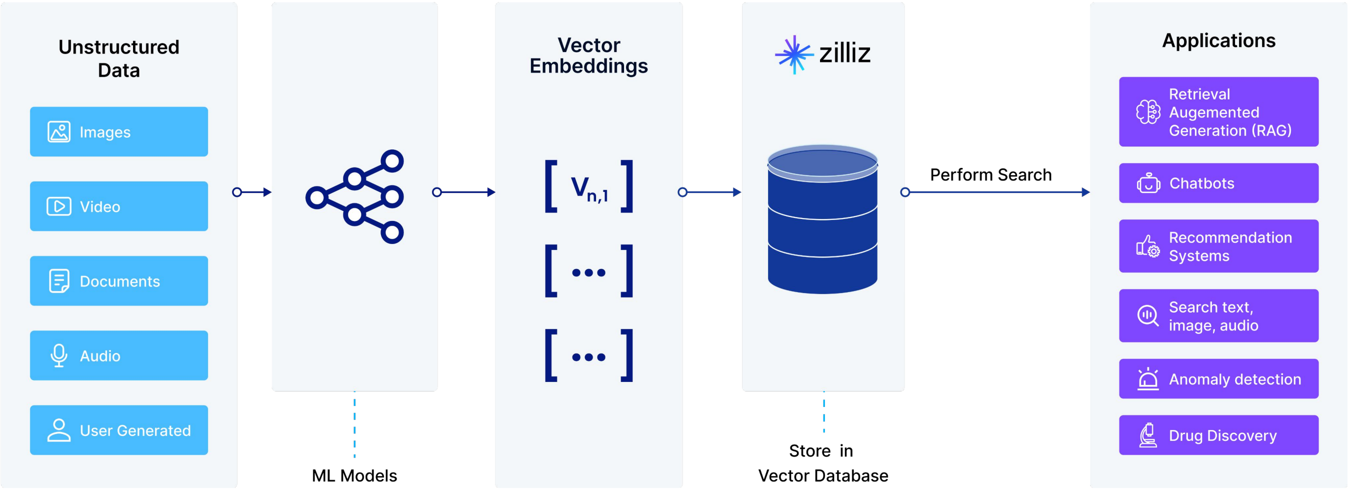


Figure 1: Managing unstructured data with vector databases

Over the years, several types of vector data management systems have emerged to meet these demands. In 2017, Meta open-sourced FAISS, dramatically reducing the cost and barriers for performing vector searches. In 2019, Zilliz introduced Milvus, a purpose-built open-source vector database that has since become a leader in the industry. This trend gained further momentum in 2022, with the rapid adoption of large language models (LLMs) like ChatGPT and the expansion of vector search capabilities into traditional database platforms such as Elasticsearch and MongoDB.

In general, vector management systems fall into four key categories:

- **Vector search libraries**, such as FAISS, Annoy, and HNSWlib. They are suitable for experimenting with vector searches or small-scale projects.
- **Purpose-built vector databases**, including Milvus and its managed cloud service, Zilliz Cloud, are perfect for scalable enterprise-level applications.
- **Lightweight vector databases**, such as Chroma and Milvus Lite. They are suitable for prototyping or experimenting with applications with a small amount of data.
- **Traditional databases with vector search add-ons**, like Elasticsearch, can handle smaller-scale vector searches.

Each category plays a distinct role in managing the complexities of unstructured data, with solutions tailored to different use cases and performance needs.

What Makes Milvus Different?

Many vector data management solutions, including vector search libraries and traditional databases with vector search add-ons, face performance bottlenecks when handling the large, complex vectors needed for machine learning tasks. These systems also lack the flexibility required for diverse machine-learning applications.

Milvus is purpose-built to manage billion-scale vector data. It is an open-source project under the Linux Foundation's Data & AI umbrella and has become the go-to solution for advanced vector search and AI applications.

Unlike traditional systems that rely on inverted indexes for basic full-text searches, Milvus enables sophisticated querying of vector data. It supports multi-vector queries, hybrid sparse and dense searches, multimodal search, and attribute filtering, making it ideal for AI and data science use cases. Whether for static or dynamic datasets, Milvus can efficiently handle real-time updates while processing queries, ensuring optimal performance even in highly dynamic environments.

Milvus is designed for scalability, distributing data across multiple nodes to ensure both high availability and superior performance. Its ability to support diverse applications—ranging from image processing, computer vision, natural language processing, and voice recognition to recommender systems and drug discovery—makes it a versatile tool for various industries. The platform also offers multiple interfaces, including RESTful APIs and SDKs in Python, Java, Go, and C++, ensuring ease of integration into different development environments.

Performance tests have shown that Milvus is up to **10 times faster** than competing systems. Additionally, as an open-source project, Milvus benefits from global contributions, resulting in a rich and continuously evolving feature set. This strong community support, combined with its superior performance and extensive functionality, has led to its widespread adoption, with hundreds of organizations worldwide relying on Milvus for their vector data management needs—and that number continues to grow.

How Does Milvus Work?

Milvus consists of a storage layer and a compute layer, and to enhance elasticity and flexibility, all components in Milvus are stateless. The system comprises four levels:

- **Access layer** — The access layer comprises a group of stateless proxies and serves as the front layer of the system and endpoint to users.
- **Coordinator service** — The coordinator service assigns tasks to the worker nodes and functions as the system's brain.
- **Worker nodes**—The worker nodes function as arms and legs. They are dumb executors that follow instructions from the coordinator service and execute user-triggered DML/DDL commands.
- **Storage** — Storage is the bone of the system and is responsible for data persistence. It comprises meta storage, log broker, and object storage.

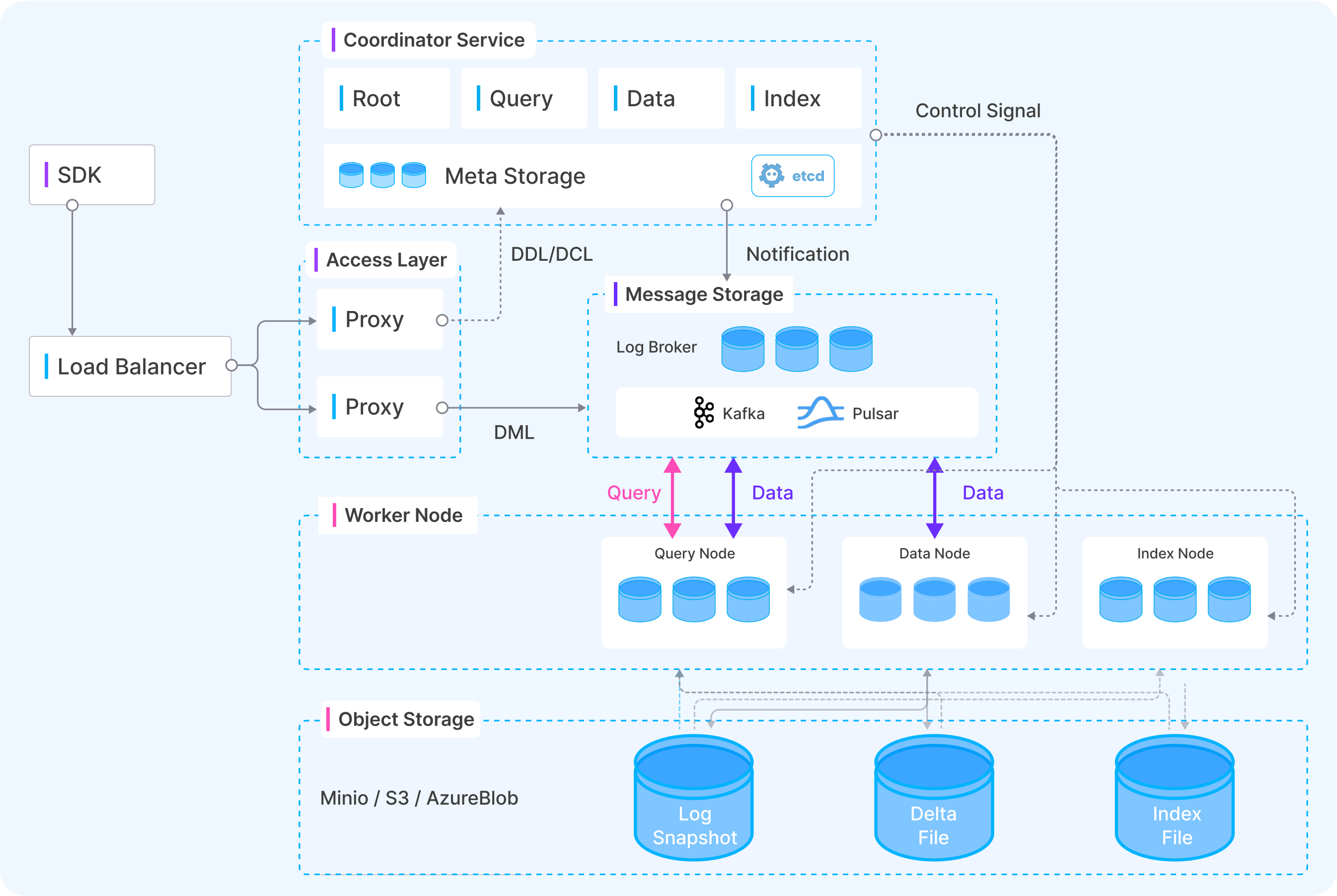


Figure 2: An overview of the Milvus architecture

The Milvus Ecosystem and Integration Capabilities

Milvus offers a robust ecosystem and extensive integration options explicitly designed for AI and machine learning applications, making it a versatile solution for managing vector data.

- **PyMilvus**: The official Python SDK for Milvus, PyMilvus simplifies the preparation and optimization of vector data for information retrieval. It supports mainstream embedding and reranking models, seamlessly integrating machine learning workflows.
- **Knowhere**: Serving as the core vector execution engine of Milvus, Knowhere integrates multiple vector similarity search libraries, including FAISS, HNSWlib, and Annoy. It supports heterogeneous computing, giving users control over whether tasks like index building and search requests are executed on CPUs or GPUs, depending on performance needs.
- **Vector Transport Service (VTS)**: Built on top of Apache SeaTunnel, VTS is an open-source tool that facilitates the movement of vectors and unstructured data into Milvus, streamlining the data transfer process for large-scale applications.
- **Attu**: This open-source management tool provides an intuitive GUI for interacting with your Milvus databases. With just a few clicks, you can visualize cluster status, manage metadata, and perform data queries, making database management more accessible.

In addition to these core tools, Milvus supports a wide range of integrations and solutions:

- **Data backup, debugging tools,** and **change data capture (CDC)** capabilities.
- **Monitoring solutions** integrated with Prometheus and Grafana, offering real-time insights into system performance.
- **Connectors** for platforms like Apache Spark, enabling efficient data migration and processing across different environments.

Furthermore, Milvus integrates with popular AI tools like LangChain and Cohere, solidifying its role as a core component in AI and machine learning workflows. These integrations make it easier to incorporate Milvus into advanced AI pipelines for tasks like natural language processing and retrieval-augmented generation.

Milvus vs. Vector Search Libraries

Vector search libraries like FAISS, Annoy, and HNSWlib are widely used for managing vector data due to their efficiency in performing similarity searches. However, these libraries are algorithms, not complete systems for managing large-scale vector data. They can run real-time queries on static datasets but lack key features required for modern AI and machine learning applications.

For instance, these libraries:

- Cannot store or manage large datasets, limiting their scalability.
- Work best only for static data, with no native support for **dynamic data** updates like insertions or deletions.
- Do not support complex queries that involve attribute filtering, multimodal data, or hybrid search.

In contrast, Milvus is a distributed, cloud-native vector database designed to handle billion-scale datasets. It offers high scalability and availability, making it an enterprise-level solution for managing and querying large volumes of vector data. Milvus also supports real-time data updates and advanced query features, making it ideal for machine learning applications requiring dynamic and complex similarity searches.

Milvus vs. Vector Search Plugins like Elasticsearch

With the rise of large language models (LLMs), many traditional databases have integrated vector search as an add-on feature to meet the growing demand for similarity search. Elasticsearch is one example of this trend. While both Elasticsearch and Milvus enable users to search for similar items across large datasets, their architectures and capabilities are fundamentally different, making them suited to distinct use cases.

Elasticsearch is built on a traditional reverse index architecture, originally designed for full-text search. Elasticsearch uses a KNN (k-nearest neighbor) plugin to accommodate vector search, which stores vectors in segments within the Lucene index. While this allows Elasticsearch to support vector search, its design is not optimized for large-scale vector management or high-dimensional similarity searches.

On the other hand, Milvus is purpose-built for vector data management. It stores, indexes, and manages massive embedding vectors generated by deep neural networks and other machine learning models. Milvus also provides advanced features like multi-vector queries, hybrid sparse and dense search, and multimodal search, which are not natively supported in Elasticsearch.

Milvus vs. Milvus Lite

In the early stages of developing AI applications, speed and flexibility often take priority over performance and stability. This is where you can choose a lightweight vector database to experiment with.

Milvus Lite is a lightweight version of Milvus designed for rapid prototyping and experimentation. It enables developers to quickly build and test end-to-end functionality in environments like notebooks without the need for a full-fledged vector database. With Milvus Lite, you can focus on running lightweight experiments, validating the effectiveness of your models, and iterating quickly—all within a streamlined setup that minimizes infrastructure overhead.

About Zilliz

Zilliz was built by the same engineers and scientists who created Milvus. It helps companies create artificial intelligence and machine learning applications more easily by managing data infrastructure. It builds database management and search technologies so users can gain the powerful insights AI brings. Zilliz Cloud is a cloud-native vector database built on Milvus. It can easily integrate with OpenAI, Cohere, HuggingFace, and other popular vectorization models. Zilliz Cloud is purpose-built and can store, index, and search through billions of vectors. It allows companies to build applications for scale and can power enterprise-grade similarity search, recommender systems, anomaly detection, and more.

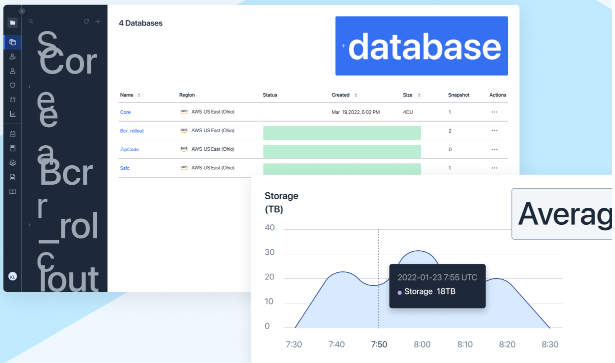
Documentation, Downloads, and Guide

- Milvus Performance Report
- Explore the Gen AI Learning Hub
- Check out the Milvus Github repository or download it (docker pull milvusdb/milvus)
- Learn more by reading the documentation

Try Zilliz for Free

Deploy a large-scale Milvus similarity search service with Zilliz Cloud in just a few minutes.

Try Zilliz Cloud for free >



4 Databases

+ database

Name	Region	Status	Created	Size	Snapshot	Actions
Core	AWS US East (Ohio)		Mar 19,2022, 6:02 PM	4CU	1	···
Bcr_rollout	AWS US East (Ohio)				2	···
ZipCode	AWS US East (Ohio)				0	···
Ssfc	AWS US East (Ohio)				1	···

Contact Us

To follow the latest updates, or if you have any questions, please feel free to contact us via:

Milvus

https://milvus.io/

milvusio.slack.com

@milvusio

https://www.youtube.com/c/MilvusVectorDatabase

https://www.linkedin.com/products/zilliz-milvus/

Zilliz

info@zilliz.com

https://zilliz.com/

@zilliz_universe

https://www.linkedin.com/company/zilliz/