zilliz

Why Vector Databases Matter for Unstructured Data

TECHNICAL PAPER

Release Date

March 28th, 2023

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 experiencing massive growth today. In addition, smartphones, IoT devices, and social media contribute to the rapid rise of unstructured data. IDC predicts that 80% 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 is data without a predefined organization system or data model. It can include text, images, videos, audio files, and more and can provide organizations with beneficial insights about their business. Yet, according to a 2019 survey by Deloitte, only 18% of organizations reported being able to take advantage of such data. The problem associated with unstructured data has never been its rarity. Instead, it's been the need for more tools and technologies to extract business value from this diverse and disordered digital resource. Unfortunately, the daunting volumes of unstructured data have discouraged companies from even attempting to mine it for nuggets of helpful information. Furthermore, due to its irregularities, traditional programs and databases need help to process unstructured data.

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

Fortunately, there are ways to gain insights into unstructured data using modern AI and ML tools. For example, machine learning models can transform unstructured data into feature vectors. Feature vectors are vectors used in machine learning to represent objects numerically. Once an object like an image or a piece of text is represented by a vector, you can use mathematical functions to analyze it. For example, you can find the distance between two feature vectors to compare how similar they are. The features that make up a feature vector can be characteristics like image color, image edges, sound lengths, sound volume, text structure, time of purchase, and much more depending on what the vector represents. These vectors can have extremely high dimensions and need to be stored in databases that are built to handle that. General databases are unable to handle the sizes of the vectors used to represent unstructured data, and they struggle with the frequent reads and writes needed to work with dynamic unstructured data.

Embeddings

In recent years machine learning technology has made it possible to transform unstructured data into vectors. A current popular way of doing this is with vector embedding. Embedding is useful for recommender systems and can convert an item, such as a word or document, into a vector. Algorithms like item2vec, word2vec, doc2vec, and graph2vec work by transforming the specified items into vectors so users can ask questions of data, such as finding similar vectors to provide recommendations. Images and text are easily represented with vectors and naturally fit this work. One example is that YouTube uses embedding to recommend videos by modeling a video as a vector and recommending similar vectors.

Processing Unstructured Data

Companies that provide recommendations have enormous amounts of data they need to work with in real time. For example, Youtube provides real-time recommendations while uploading 500 hours of new videos per minute. Their recommendation algorithm is more complicated than a simple vector similarity search; complex queries use multiple vectors to filter data and ask questions about it. Many companies working with unstructured data vectorize it and then use machine learning techniques to analyze it. Companies need scalable vector management systems to be able to ask complex questions of unstructured data.

Popular use cases for using Unstructured Data

Natural language processing — Q&A 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 it interpret user messages to 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 for matching keywords, 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 are supposed to look a certain way, it would be very tedious and prone to error for a person to check each one individually manually. But once each image has been turned into a feature vector, an analyst can write a program to mathematically check for unusual cases.

The unstructured data workload

Unstructured data is complicated, and the vectors representing it are huge to capture that complexity. So when you're working with unstructured data, it's easy to find yourself needing to handle billions of vectors, and you need databases that can handle that. And therefore, there is a pressing need for a scalable vector data management system that can support fast query results on massive datasets and efficiently handle insertions and deletions for dynamic data.

What makes Milvus different?

Current popular systems for managing vector data run into performance problems when they attempt to handle the large and complicated vectors often required for machine learning analysis. They also need to be more flexible to work with the many different kinds of applications machine learning is used for.

Milvus is a purpose-built vector database designed to handle large-scale vector data. It allows users to ask advanced queries of vector data rather than limiting them to simple things like full-text searches relying on an inverted index. It can also quickly update data while queries are processed, allowing for dynamic data. Milvus distributes data across multiple nodes and is scalable and available. It outperforms competitors because it can support multi-vector queries and attribute filtering, which makes it suitable for AI and data science purposes. It has many application interfaces, including RESTful APIs and SDKs in Python, Java, GO, and C++. Companies in various fields use Milvus, including image processing, computer vision, natural language processing, voice recognition, recommender systems, and drug discovery.

Milvus is a purpose-built vector database designed to handle large-scale vector data. It allows users to ask advanced queries of vector data rather than limiting them to simple things like full-text searches relying on an inverted index. It can also quickly update data while queries are processed, allowing for dynamic data. Milvus distributes data across multiple nodes and is scalable and available. It outperforms competitors because it can support multi-vector queries and attribute filtering, which makes it suitable for AI and data science purposes. It has many application interfaces, including RESTful APIs and SDKs in Python, Java, GO, and C++. Companies in various fields use Milvus, including image processing, computer vision, natural language processing, voice recognition, recommender systems, and drug discovery.

The Milvus stack

The Milvus platform consists of Milvus, Towhee, Knowhere, Py-Milvus, and Attu, which comprise the Milvus Stack.

Milvus is the world's most-popular vector database. It's open source and a Linux Foundation Project for Data & AI graduate project.

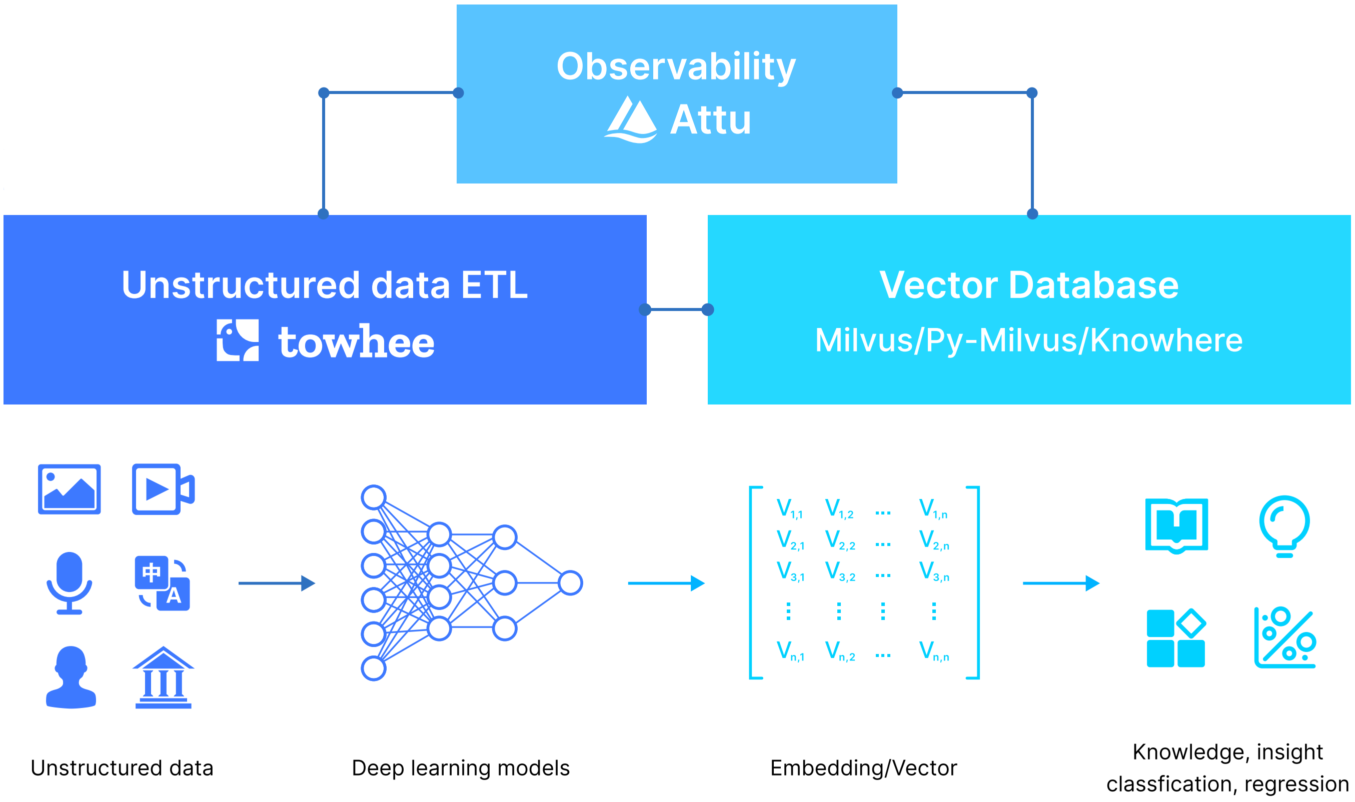
Towhee is an open-source resource for building data processing pipelines for AI applications using both neural network models and traditional methods. It provides models, algorithms, and transformations to build pipelines and has a Pythonic API. In addition, it supports data processing for unstructured data, including images, videos, text, and more.

Knowhere is the core vector execution engine of Milvus which incorporates several vector similarity search libraries, including Faiss, Hnswlib, and Annoy. Knowhere supports heterogeneous computing and controls which hardware (CPU or GPU) to execute index building and search requests.

Py-Milvus is a Python SDK of Milvus.

Finally, **Attu** is an open-source management tool for Milvus with an intuitive GUI, allowing you to interact easily with your databases. With just a few clicks, you can visualize your cluster status, manage metadata, perform data queries, and much more.

Together Milvus, Towhee, Knowhere, Py-Milvus, and Attu allow users to work with large sets of complex unstructured data and solve the challenges that often come up with this kind of work.



Observability
Attu
Unstructured data ETL
towhee
Vector Database
Milvus/Py-Milvus/Knowhere
中
A
$V_{1,1}$ $V_{1,2}$ ... $V_{1,n}$
$V_{2,1}$ $V_{2,2}$ ... $V_{2,n}$
$V_{3,1}$ $V_{3,2}$ ... $V_{3,n}$
$V_{n,1}$ $V_{n,2}$ ... $V_{n,n}$
Unstructured data
Deep learning models
Embedding/Vector
Knowledge, insight
classfication, regression

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 of 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 and 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.

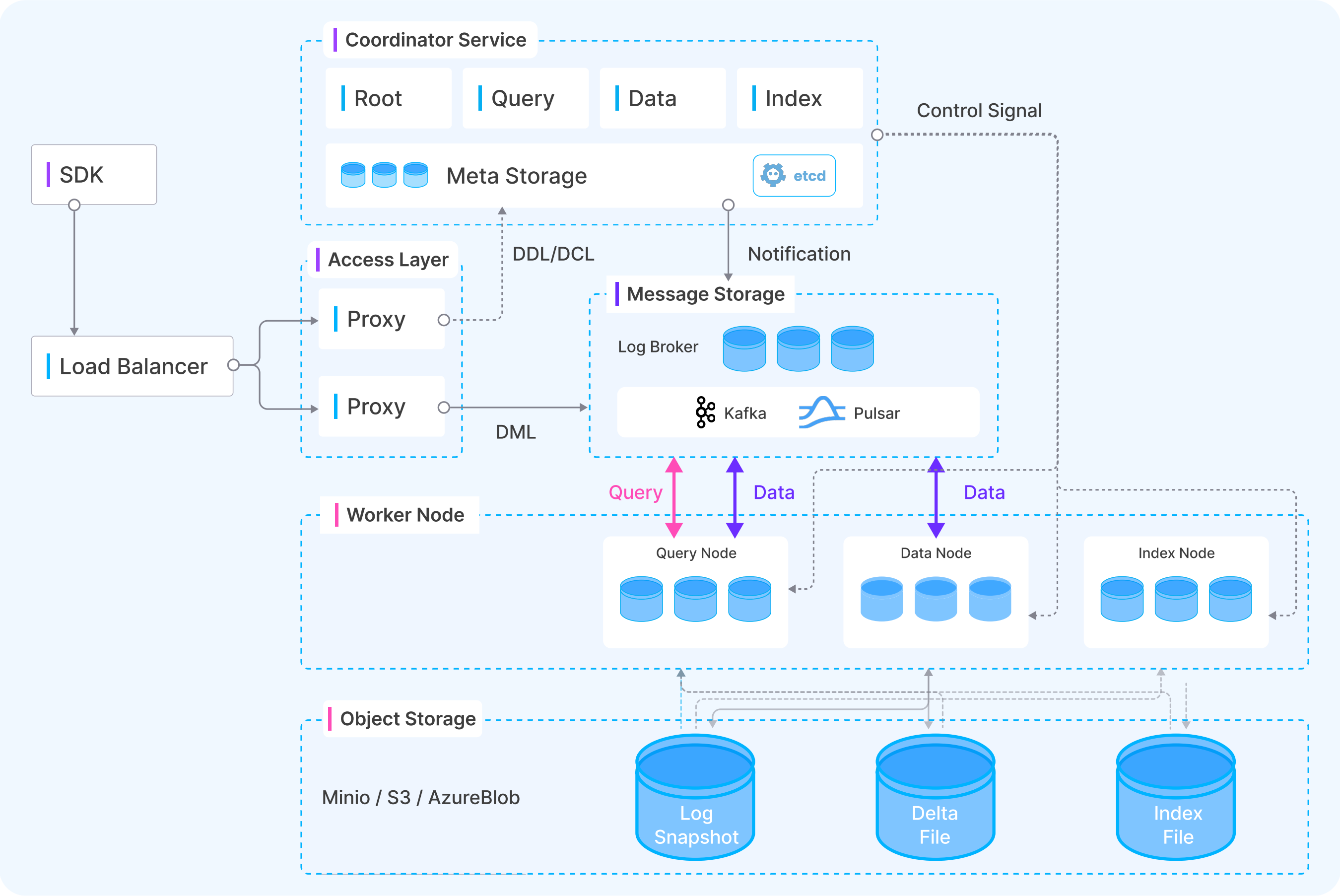


Figure 1. An overview of the Milvus architecture

Milvus vs. Elasticsearch

Milvus and Elasticsearch are two of the most popular similarity search engines. While both help users find similar items within a large dataset, some critical differences between them make them better suited for specific applications.

Elasticsearch uses the reverse index and builds vector search capability on top of the existing search architecture. It also has a KNN plugin that stores vectors separately in each segment in the Lucene index. In contrast, Milvus stores, indexes, and manages massive embedding vectors generated by deep neural networks and other machine learning (ML) models.

These differences have implications for performance and scalability. For example, Elasticsearch is built on top of the existing search architecture, which gives it scalability; it can handle many queries on a large dataset. On the other hand, Milvus is designed for handling large-scale vector similarity search tasks, making it suitable for machine learning applications requiring similarity search.

Milvus vs. other databases

PGvector is a powerful open-source tool that enables users to search for highly similar vectors in Postgres. One of the key benefits is that it is based on the popular Postgres database. Another benefit is that it can handle tables larger than the 32 TB limit imposed by Postgres on non-partitioned tables. Partitioning tables with PGvector allows you to create thousands of partitions, each up to 32 TB in size. Additionally, PGvector supports replication through the write-ahead log (WAL) feature, essential for achieving point-in-time recovery in the event of a failure.

For large tables with 2,000 or more vectors, it is recommended to use dimensionality reduction or compile Postgres with a larger block size and edit the default limit. Doing so can help improve the efficiency of your searches and ensure that you are within the limits of your system.

Milvus vs. other vector solutions

Facebook Faiss and Microsoft SPTAG are other ways of working with vector data, but they are algorithms, not complete systems for managing vector data. They can't store large amounts of data, only work well for real-time queries with static data (not dynamic data), and don't support complex queries.

Alibaba AlnalyticDB-V and Alibaba PASE support vector similarity search, but they aren't purpose-built for vector data. They're relational databases with an additional table column that can store vectors. Because of this, they're not optimized for vector data and process it inefficiently. They also don't support advanced queries that involve complex questions or multiple vectors.

Vearch is a system designed for vector search. It is purpose-built but it only works well with small volumes of data. Vearch also doesn't support advanced multi-vector queries.

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 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, Cohee, 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.

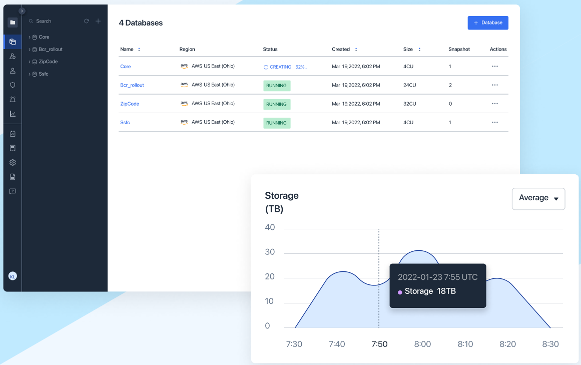
Documentation, Downloads, and Guide

- Milvus Performance Evaluation 2023
- 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

Name	Region	Status	Created	Size	Snapshot	Actions
Core	AWS US East (Ohio)	CREATING 52%...	Mar 19,2022, 6:02 PM	4CU	1	···
Bcr_rollout	AWS US East (Ohio)	RUNNING	Mar 19,2022, 6:02 PM	24CU	2	···
ZipCode	AWS US East (Ohio)	RUNNING	Mar 19,2022, 6:02 PM	32CU	0	···
Ssfc	AWS US East (Ohio)	RUNNING	Mar 19,2022, 6:02 PM	4CU	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/