ClickHouse meets LLMs

Vector Search at Scale without a “Vector DB”

Jason Wong (Senior Support Engineer)

ClickHouse

ClickHouse is not “just OLAP”

it can be a **high-performance vector retrieval engine** in LLM systems.

Everyone talks about LLMs.
The real bottleneck is **retrieval at scale**.

- LLM magic is backed by RAG

- A successful RAG needs to…
  - Maintain high quality of embedding
  - Minimal retrieval latency
  - Holding an optimized storage

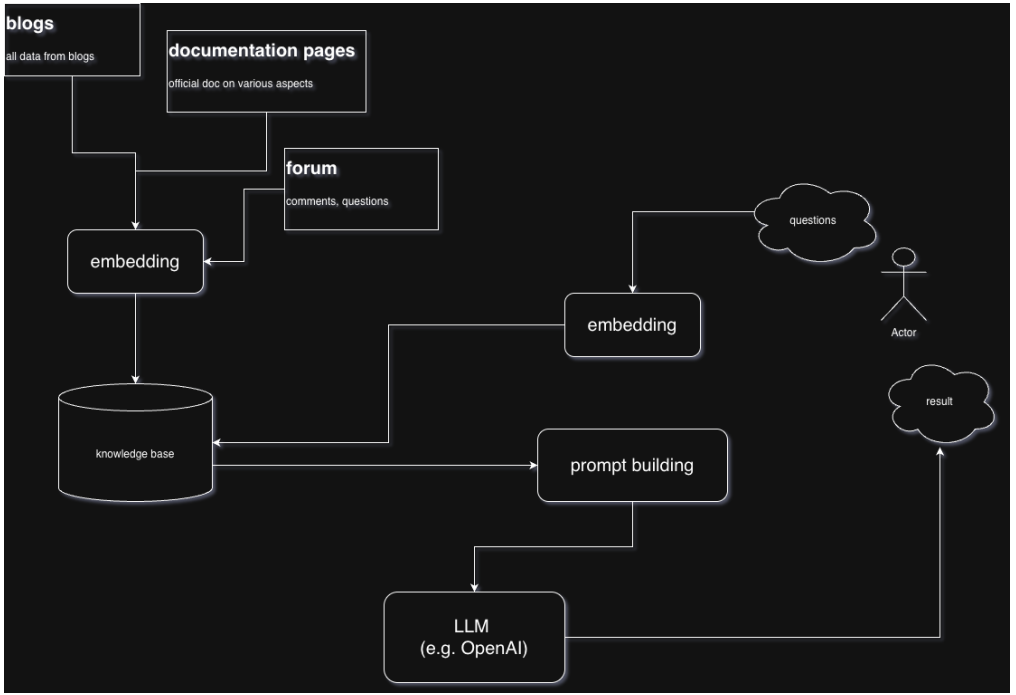
- Which product support all these????

LLM: is a sophisticated type of artificial intelligence (AI) program that is trained on a massive amount of text data to understand, process, and generate human-like language

RAG: is an artificial intelligence (AI) technique that enhances the capabilities of large language models (LLMs) by connecting them to external knowledge sources

Typical LLM Architecture

- Vector DB often introduced as yet another system
- Operational overhead
- Cost + data duplication



Where ClickHouse Fits

- 1st class support for vectors (aka Arrays)

- Super fast ingestion and retrieval of data to provide real-time response

- ANN support (HNSW at the moment)

- Metadata filtering + vector search together

ANN: a computational model in artificial intelligence (AI) and machine learning (ML) that is inspired by the structure and function of the human brain's biological neural networks

HNSW: refers to the use of the Hierarchical Navigable Small World (HNSW) algorithm for performing Approximate Nearest Neighbor (ANN) search, a key technology in modern AI and machine learning applications

Where ClickHouse Fits

- ClickHouse schema for the vector column is simply Arrays of Float32

- No special or dedicated data type needed - removing unnecessary type conversions for ingestion and retrieval

```
CREATE TABLE playground.docs
(
    doc_id       UInt64,
    chunk_id     UInt32,
    content      String,
    embedding    Array(Float32),
    source       LowCardinality(String),
    tenant_id    UInt32,
    ts           DateTime
)
ENGINE = MergeTree
ORDER BY (tenant_id, doc_id, chunk_id);
```


Where ClickHouse Fits

- ClickHouse can filter meta-data which is not vector based (e.g. tenant_id and source), not all vector-db support

- It also run KNN distance functions based on a vector column

- the BEST of both worlds

```
SELECT
    content,
    cosineDistance(embedding, [
        -0.0066272463, -0.07342478, -0.1604181, 0.031773847,
        -0.048222084, -0.023915742, 0.038059756, -0.017205033,
        -0.017487692, 0
    ]) AS score
FROM playground.docs
WHERE
    tenant_id = 88
    AND source = 'meetup_workshop'
    AND ts >= now() - INTERVAL 30 DAY
ORDER BY score asc
LIMIT 5;
```


Embedding Generation

Knowledge-base and user questions

- Language: Python 3

- Library / Module
    - sentence-transformers (Hugging Face)
    - clickhouse-connect (ClickHouse)

Knowledge base Retrievals

- No SECRET at all, just a normal Query using ClickHouse built-in functions

Takeaways

- One system (OLAP + Vector DB)

- Cost efficiency
    - columnar store
    - ZSTD compression

- Queries supporting meta-data and vector filtering

About ME(me)

- Senior Support Engineer (ClickHouse)
- ElasticSearch Expert (full-text search, inverted-index…)
- Rust / Go / Vue.js Developer

LinkedIn

Demo Source code

Thank you!