

向量数据库选型评估






























业务价值

- 1. 为 LLM 存储长期记忆
- 2. 解决 LLM 输入的 Token 限制：不会在每个 LLM 提示中都发送所有文档，而是只发送少数相关度最高的文档
- 3. 私有部署解决数据安全问题

核心功能

高效地存储、对比和检索海量的向量

Round 0：确定向量数据库的范围

向量数据库	LangChain 支持 [Link]	Semantic Kernel [Link]	LlamaIndex [Link]	OpenAI 支持 [Link]
AnalyticDB				
Annoy				
Atlas				
AwaDB				
Azure Search				
BagelDB				
Cassandra				
Chroma				
Clarifai				
ClickHouse Vector Search				
Activeloop's Deep Lake				
Dingo				
DocArrayHnswSearch				
DocArrayInMemory Search				
ElasticSearch				
FAISS				
Hologres				
LanceDB				

Marqo	✓			
MatchingEngine	✓			
Meilisearch	✓			
Milvus	✓	✓	✓	✓
MongoDB Atlas	✓		✓	
MyScale	✓		✓	
OpenSearch	✓		✓	
pg_embedding	✓			
PGVector	✓	✓		✓
Pinecone	✓	✓	✓	✓
Qdrant	✓	✓	✓	✓
Redis	✓	✓	✓	✓
Rockset	✓			
ScaNN	✓			
SingleStoreDB	✓			
scikit-learn	✓			
StarRocks	✓			
Supabase (Postgres)	✓		✓	✓
Tair	✓		✓	
Tigris	✓			
Typesense	✓			
USearch	✓			
Vectara	✓			
Weaviate	✓	✓	✓	✓
Xata	✓			
Zilliz	✓			✓

选择在 LangChain、Semantic Kernel、LlamaIndex 和 OpenAI 中支持较多的向量数据库

Round 1：根据类型筛选 ✓

向量数据库	数据库类型	开源	部署	结论
AnalyticDB	扩展向量模块 ✗	否 ✗	Managed ✗	✗
Azure Search		否 ✗	Managed ✗	✗
Chroma	纯向量库	是	Memory ✗	✗

ElasticSearch / OpenSearch	扩展向量模块 ✗	是	Managed Self-Hosted	✗
Milvus	纯向量库	是	Managed Self-Hosted	✓
PGVector	扩展向量模块 ✗	是	Self-Hosted	✗
Pinecone	纯向量库	否 ✗	Managed ✗	✗
Qdrant	纯向量库	是	Managed Self-Hosted	✓
Redis	扩展向量模块 ✗	是	Self-Hosted	✗
Supabase (Postgres)	扩展向量模块 ✗	是	Self-Hosted	✗
Weaviate	纯向量库	是	Managed Self-Hosted	✓
Zilliz (Milvus)	纯向量库	是	Managed ✗	✗

以下类型的向量数据库不考虑:

- 不开源
- 不能本地部署
- 不能持久化的，只能 Memory 存储的
- 其他类型数据库（OLTP、NoSQL、OLAP）扩展出向量模块的不选择：这些数据库大多数都没有对向量的使用进行优化。另外从微服务的角度说，每个专业的服务（or 工具）应该专注于解决一个的特定事情，能保持服务进行独立的扩展、部署和维护

Round 2：文献调研

	调研项	Weaviate	Qdrant	Milvus
社区	开源许可证	BSD-3-Clause	Apache 2.0	Apache 2.0
	商业公司维护	是	是	是
	发布时间	2019	2021	2019
	Star ★	7.1k	12.2k	22k
	DB-Engines Ranking	5	7	4
	网址	Link	Link	Link
功能	核心特性	<ul style="list-style-type: none">支持向量与对象的存储向量索引与倒排索引组合集成 ML / Embedding 模型	<ul style="list-style-type: none">丰富的数据类型Payload 可以索引WAL 日志写入SIMD	<ul style="list-style-type: none">极高的检索性能非结构化数据的极简管理跨平台实时检索和分析

架构		<ul style="list-style-type: none">支持推荐、总结等能力		<ul style="list-style-type: none">具有很高的容灾与故障转移能力高度可拓展
	数据量级			10 亿级
	Index Types ★	<ul style="list-style-type: none">倒排索引向量索引<ul style="list-style-type: none">HNSW	<ul style="list-style-type: none">Payload 索引全文索引向量索引<ul style="list-style-type: none">HNSW	<ul style="list-style-type: none">标量索引向量索引<ul style="list-style-type: none">FLATIVS_FLATIVF_SQ8IVF_PQHNSWANNOYBIN_FLATBIN_IVF_FLAT
	Similarity Metrics ★	<ul style="list-style-type: none">Cosine SimilarityDot ProductEuclidean DistanceHammingManhattan Distance	<ul style="list-style-type: none">Cosine SimilarityDot ProductEuclidean Distance	<ul style="list-style-type: none">Euclidean DistanceInner ProductJaccardTanimotoHammingSuperstructureSubstructure
	混合查询 ★	√	√	√
	数据模型	<ul style="list-style-type: none">ClassProperty	<ul style="list-style-type: none">CollectionPointsSharding	<ul style="list-style-type: none">Collection: TableEntity: RowField: Field (Value or Vector)Primary KeySharding / Partition / Segment
	Backup & Restore ★	√	√ Snapshots	√ Binlog、日志持久化、日志快照
	编程接口	<ul style="list-style-type: none">RESTfulGraphQLPythonJavaGoJavascript	<ul style="list-style-type: none">RESTfulOpenAPI 3.0gRPCPythonGoRustJavascript	<ul style="list-style-type: none">RESTfulJavaGoJavascript
	RBAC	×	×	√
架构	编程语言	Go	Rust	Go
	架构图			

				
架构复杂度 ★	简单	简单	复杂，包含组件有： <ul style="list-style-type: none">Access LayerCoordinator ServiceWorker Node，细分为：<ul style="list-style-type: none">data nodequery nodeindex nodeStorage，细分为：<ul style="list-style-type: none">元数据存储（meta store）消息存储（log broker）对象存储（object storage）	
分布式架构	√	√	√ 分布式 MPP 架构	
存算分离	×	×	√	
Replication ★	√ (Class-level replication)	√ (Collection-level replication)	√	
Sharding ★	√ 静态数据 Sharding	√ 静态数据 Sharding	√ 动态 Segment 替换	
Partition ★	×	×	√	
可扩展性	<ul style="list-style-type: none">Resharding 复杂且耗时	<ul style="list-style-type: none">Resharding 复杂且耗时	<ul style="list-style-type: none">存算分离，节点无状态，扩缩容简单动态扩缩容水平扩展	
数据一致性	<ul style="list-style-type: none">Eventual ConsistencyTunable consistency	<ul style="list-style-type: none">Eventual ConsistencyTunable consistency	<ul style="list-style-type: none">Eventual ConsistencyImmediate ConsistencySession ConsistencyTunable Consistency	
部署	服务器成本 ★	低	低	高
	组件依赖 ★	内置组件： <ul style="list-style-type: none">系统进程	内置组件： <ul style="list-style-type: none">系统进程 外部组件：	内置组件： <ul style="list-style-type: none">ProxyRoot coord

	<ul style="list-style-type: none">• module service (可选) 外部组件： <ul style="list-style-type: none">• Kubernetes	<ul style="list-style-type: none">• Kubernetes	<ul style="list-style-type: none">• Query coord• Index coord• Data coord• Query node• Index node• Data node 外部组件： <ul style="list-style-type: none">• Kubernetes• etcd：元数据存储• MinIO：对象存储• Kafka/Pulsar: 对象存储
云原生	√	√	√

关键评估指标：

- 社区
 - Star
- 功能
 - Index Types
 - Similarity Metrics
 - Backup & Restore
- 架构
 - 架构复杂度
 - Replication
 - Sharding
 - Partition
- 部署
 - 服务器成本
 - 部署组件

术语解释：

- Faiss 是由 Facebook 开发的适用于稠密向量匹配的开源库，支持 C++ 与 Python 调用。Faiss 支持多种向量检索方式，包括内积、欧氏距离等，同时支持精确检索与模糊搜索
- ANN：近似最近邻搜索，ANN 算法会预先构建索引，通常，有 3 种索引结构：图索引、树索引、哈希索引。选择不同的索引算法会影响搜索速度、内存使用情况和准确性。各种类型的 ANN 索引算法主要分为 2 种思路：缩小搜索范围和将高维向量空间分解为低维子空间。
- HNSW：Hierarchical Navigable Small World，图索引算法，通过创建多层接近图（Proximity graph）来索引向量空间。
- ANNOH：Approximate Nearest Neighbor Oh Yeah，树索引算法，通过构建二叉树森林来索引向量空间。ANNOY 使用在向量空间中距离两个随机向量相等的超平面来将空间分成 2 个子空间。对每个子空间重复此过程，直到每个子空间中最多有 K 项。
- IVF: Inverted File Index，IVF 通过将整个 vector 空间拆分成 k 个子空间，并对每个子空间找到一个代表质心（centroid）。并将子空间的所有 vector points 都 match 到这个质心上。给定一个要查询的 vector，先通过和所有的质心比较找到最近的质心，然后在这个质心所代表的子空间里搜索最近的点

- Jaccard/Tanimoto：是衡量两个二进制向量之间的重叠量
- Hamming：是对两个二进制向量中不同的向量元素数量的统计
- Partition 的意义在于通过划定分区减少数据读取，而 Sharding 的意义在于多台机器上并行写入操作。
- 静态数据 Sharding：采用静态数据分片后，如果数据规模超过服务器存储上限，您需要为集群添加更多机器并重新对数据进行分片。这个过程复杂且耗时。而且，数据分片不均衡可导致性能瓶颈，降低系统效率。
- 动态 Segment 替换：动态分配节点有益于实现更轻松的扩展和更合理的资源分配，从而确保系统延迟和吞吐量。

Round 3：试用评估

	向量数据库	参数	Recall	QPS	ART (ms)	T95 (ms)	T99 (ms)	Index Build(s)
最佳召回	Weaviate	<ul style="list-style-type: none">• m: 72• efCon: 512• ef: 512	1	25.8	38.5	51.5	62.5	311
	Qdrant	<ul style="list-style-type: none">• m: 72• efCon: 512• ef: 512• quantization : True• rescore: True	0.99987	89	11.5	15.2	18	118
最佳QPS	Weaviate	<ul style="list-style-type: none">• m: 72• efCon: 256• ef: 16	0.97628	114	8.5	10.2	13.5	101
	Qdrant	<ul style="list-style-type: none">• m: 72• efCon: 256• ef: 16• quantization : True• rescore: True	0.99145	231.9	4.2	4.9	5.3	55

		Weaviate	Qdrant
高可靠性	HA	<ul style="list-style-type: none">• 停止 1 节点后，表写入 $\sqrt{\quad}$• 停止 1 节点后，表读取 $\sqrt{\quad}$• 停止 1 节点后，表删除 \times• 停止 1 节点后，单副本表创建 \times• 停止 1 节点后，多副本表创建 \times	<ul style="list-style-type: none">• 停止 1 节点后，表写入 $\sqrt{\quad}$• 停止 1 节点后，表读取 $\sqrt{\quad}$• 停止 1 节点后，表删除 $\sqrt{\quad}$• 停止 1 节点后，单副本表创建 $\sqrt{\quad}$• 停止 1 节点后，多副本表创建 \times

		<ul style="list-style-type: none">• Bug 1: 停止 Following 节点，其他节点创建单副本表后，该 Following 节点不能启动• Bug 2: 停止 Following 节点，其他节点删除表后，该 Following 节点不能启动• Bug 3: 停止 Master 节点，其他节点写入表数据，启动 Master 节点后，从 Master 节点读取报错	
	容错能力	√ Replication	√ Replication
运维成本	部署复杂度	容器部署简单，裸机部署简单	容器部署简单，裸机部署复杂 (需编译和安装多个依赖)
	扩缩容复杂度	简单	简单
	压测运行资源消耗	CPU: 2 cores / MEM: 1.2G	CPU: 2 cores / MEM: 800M
	可观测性	√ (with Grafana Dashboard)	√

关键评估指标：

- 业务性能：召回率
- 技术性能：QPS / 响应时间
- 高可靠性：HA、容错能力
- 运维成本：部署复杂度、扩缩容复杂度、副本自动修复能力、数据自动均衡能力

Bug 1 重现步骤：

1. 停止 Master 节点
2. 停止 Follow 节点，启动 Master 节点
3. 用 Master Url 节点上创建单副本表
4. Follow 节点启动失败

Bug 2 重现步骤：

1. 停止 Master 节点
2. 停止 Follow 节点，启动 Master 节点
3. 用 Master Url 节点上删除表
4. Follow 节点启动失败

Bug 3 重现步骤：

1. 停止 Master 节点
2. 其他节点写入表数据，表为 3 副本
3. 启动 Master 节点
4. 从 Master 节点读取该数据报 consistency level 错误

参考文档

[What is a Vector Database?](#)

[什么是向量相似性搜索？](#)

[The Best Vector Database for Stablecog's Semantic Search](#)

[Open Source Vector Database Comparison](#)