RAG Solutions.
AI Search vs PostgreSQL pgvector in PROD

AI Search

PostgreSQL

Advantages and disadvantages
from the perspective of an engineer
who has used both solutions in a
production.

Aleksei Kolesnikov
Staff Software Engineer

Simple Guide how to choose the proper solution for your RAG.

1. Education Curve

BETTER

PostgreSQL

Well known Relational Structure.
Just need to learn about vectors and how to apply index.

WORSE

AI Search

Complex Service with API capabilities,
ingestion and migration limitations, configurations, etc.

AI Search requires examples and prototypes explaining how it works inside, how to ingest large amount of data, what modern libraries support AI Search, etc.

2. Costs

BETTER

PostgreSQL

Multiple times cheaper solution.
Significantly cheaper when the data size grows.
Supported by different underlying services such as CosmosDB and Azure SQL, and their different tiers.

WORSE

AI Search

Quite expensive.

Especially if you have a lot of ingested data.

To achieve 99.9% SLA uptime you need:

1. for read-only operations at least 2 replicas needed;
2. for read-write operations at least 3 replicas needed.

It's essential to justify to organization's high-level executives the added costs of AI search, especially when cost-effective solutions like Postgres pgvector and Mongo Atlas Vector Search are available.

3. Space, Backup, and Migration

BETTER

PostgreSQL

1. Full control on how to store the data:
   - You can easily calculate the amount of space you need to place your documents or Q&A pairs;
2. Complete and clear management of backups;
3. Transparent migration process.

WORSE

AI Search

1. Occupied space depends on ingestion mechanism details such as chunking, vector size, field types:
   - 1GB of PDF original files can occupy between 50MB to 600MB of AI Search storage space.
2. Modifying an index can require a full rebuild, involving data download, new index creation, data re-upload;
3. There are no standard tools for transferring documents between indexes. Even the code solutions come with the set of limitations.

For AI Search, a well-planned data structure is essential.
A primary decision point is to choose between a “universal index” and “separate indexes” per document type.

4. Data Ingestion

EQUAL

AI Search

Provides Pull and Push mechanisms, and import wizards.

- **Pull** documents from supported storages to a chosen index, with or without Skillset;
- **3 Import wizards:** differing import capabilities and their results across **two** Azure Portal wizards and **one** OpenAI Playground import:
- **Push:** Custom solution with REST API and KernelMemory.

EQUAL

PostgreSQL

Bulk insertion, snapshots, and personalized solutions are just a few of the many data ingestion options offered by Postgres.

While AI Search offers numerous options, **wizards** and **pull** mechanisms are primarily suited for prototyping.

For production-ready data, it’s necessary to develop a custom push mechanism.

5. Scalability

BETTER

PostgreSQL

Different scalability options are available with underlying services like Azure SQL Database and CosmosDB.

They offer easy scaling mechanisms, usually achievable in just a few clicks.

WORSE

AI Search

AI Search scales within its tier via data replicating replicas. Smooth scaling is achievable with available replicas.

Yet, transitioning between tiers necessitates a data migration process.

Scaling of AI Search has its challenges and offers less direct control.

6. Performance

BETTER

AI Search

1. Performance in AI Search largely depends on the number of replicas.
2. Different index schemas typically show stable and similar results.

WORSE

PostgreSQL

Vector performance in PostgreSQL is fairly good, though it falls short of AI Search.

Load testing indicated that performance starts to degrade with over 20-25 requests per second when more vectorized data is uploaded.

While AI Search showcases superior performance, PostgreSQL results may fit your specific use cases.

7. Search Quality

BETTER

AI Search

AI Search incorporates advanced techniques like Hybrid Search and Re-ranking, in addition to an advanced scoring system.

Also, it offers the flexibility to control how searching is conducted and how scoring algorithms are applied.

WORSE

PostgreSQL

With PGvector, you control the vector's similarity.
It supports Re-ranking via a cross-encoder and Hybrid searching using **tsvector** and **pgvector**.

However, any additional Re-ranking mechanism needs to be manually implemented.

AI Search offers marginally more flexibility through the configuration of Index schemas.

8. Integration with LLM Solutions

AI Search

AI Search can be seamlessly integrated into platforms such as SemanticKernel, LangChain, and KernelMemory.

Moreover, it supports REST API calls, offering broad integration capabilities should you encounter any limitations in support.

PostgreSQL

Postgres can be integrated into LangChain and SemanticKernel using native functions. Since you have full control over the data source, you can utilize any integration systems.

However, this might necessitate writing additional code.

AI Search demonstrates impressive integration capabilities with various languages and tools.

Other Options to store your RAG data

Elastic Search

Qdrant

MongoDB

Milvus

Options such as MongoDB, Qdrant, and Milvius have their own challenges:

1. Service Level Agreements limitations;
2. Resource management;
3. Complexities in integration.

These options may still be viable for production use, but they should be carefully evaluated during the decision-making process of your architectural design.

Aleksei Kolesnikov
Staff Software Engineer