MariaDB Vector

a new Open Source vector database that you are already familiar with

Sergei **Golubchik**

What is “Vector Search”

- Semantic text search
- Image search
- Music search
- Generative AI
  - RAG, Retrieval Augmented Generation

Hybrid, too

How?

- You convert data texts (images, audio) into vectors
- Store vectors in the database
- To search you convert the query text (image, audio) into a vector
- Use vector search!

What is “Vector”

- A list of floating point numbers
  - e.g. [0.4187, 0.8099, 0.82319, 0.5982, 0.03326]
- Typical length: 20–2000 numbers
- Search for the “nearest”
  - the search is **approximate**

Example

```sql
CREATE TABLE embeddings (
  doc_id BIGINT UNSIGNED PRIMARY KEY,
  embedding BLOB NOT NULL,
  VECTOR INDEX (embedding)
);
```

```
CREATE TABLE embeddings (
  doc_id BIGINT UNSIGNED PRIMARY KEY,
  embedding BLOB NOT NULL,
  VECTOR INDEX (embedding)
    MAX_EDGES_PER_NODE=8
    DISTANCE_FUNCTION=COSINE
);
```

```python
from openai import OpenAI
client = OpenAI()
model = "text-embedding-3-small"

def get_embedding(text):
   return client.embeddings.create(input = [text],
               model=model).data[0].embedding
```

```python
import mariadb
import array

v = get_embedding(document[i])
cur.execute("INSERT embeddings VALUES (%d, %s)",
              (i, array.array("f", v).tobytes()))
```

```
import mariadb


v = get_embedding(document[i])
cur.execute(
   "INSERT embeddings VALUES (%d, Vec_FromText(%s))",
   (i, str(v)))
```

```
import mariadb
import array

q = get_embedding(user_question)
cur.execute("""
   SELECT doc_id FROM embeddings
      ORDER BY VEC_DISTANCE_COSINE(%s, embedding)
      LIMIT 5
""", array.array("f", q).tobytes()))
```

Configuration

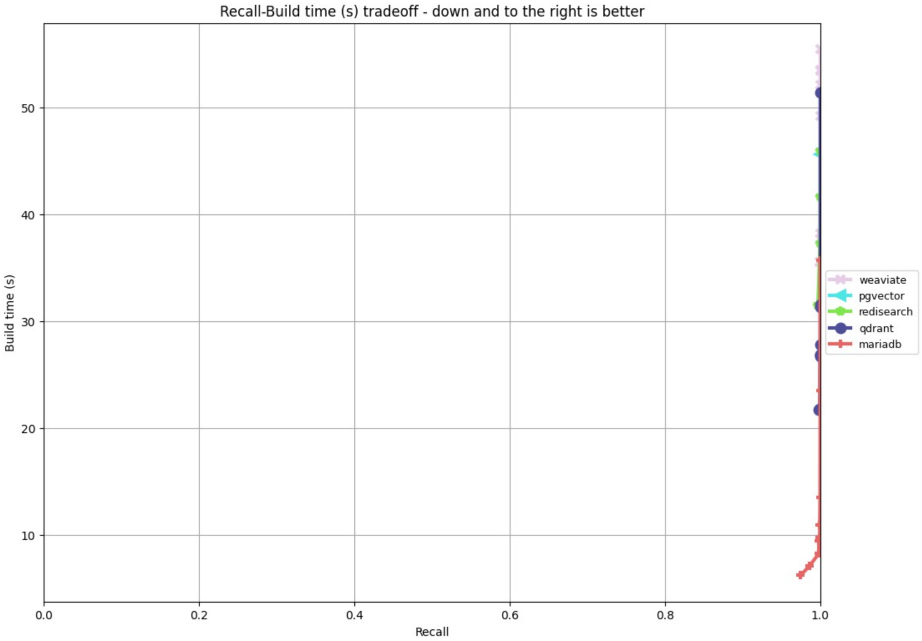
Server variables

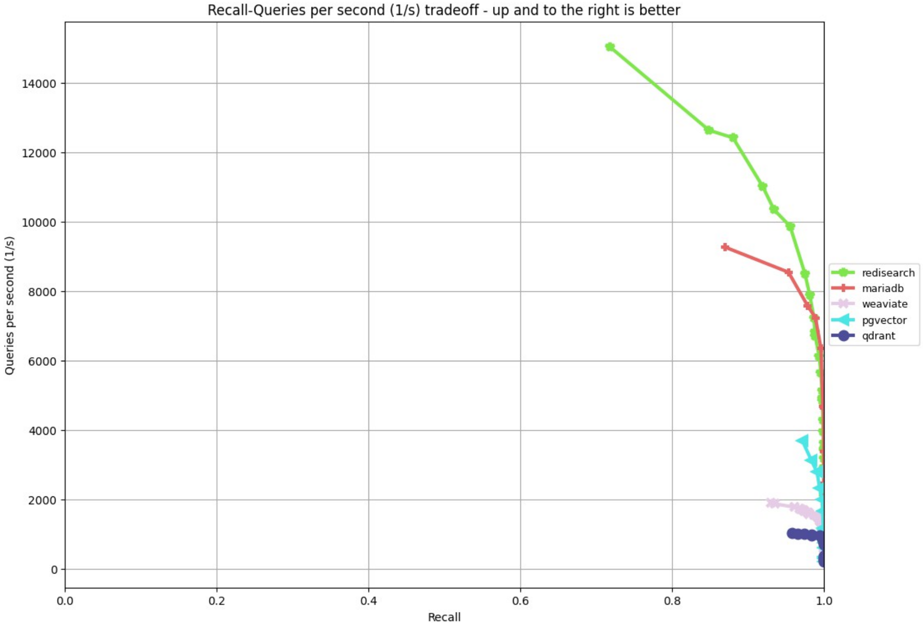
- `mhnsw_cache_size`
- `mhnsw_distance_function`

- `mhnsw_max_edges_per_node` (M)
- `mhnsw_min_limit` (ef)

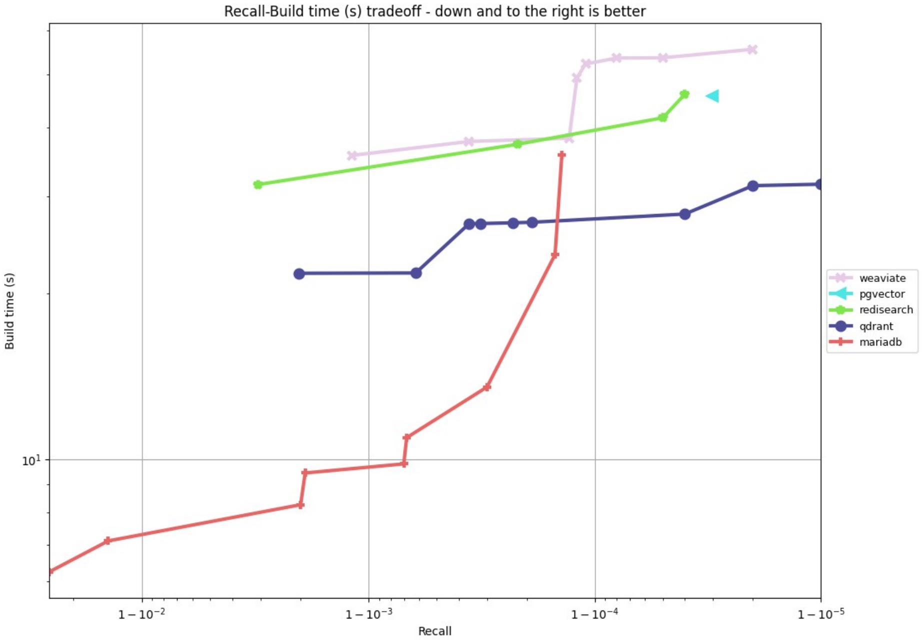
Performance

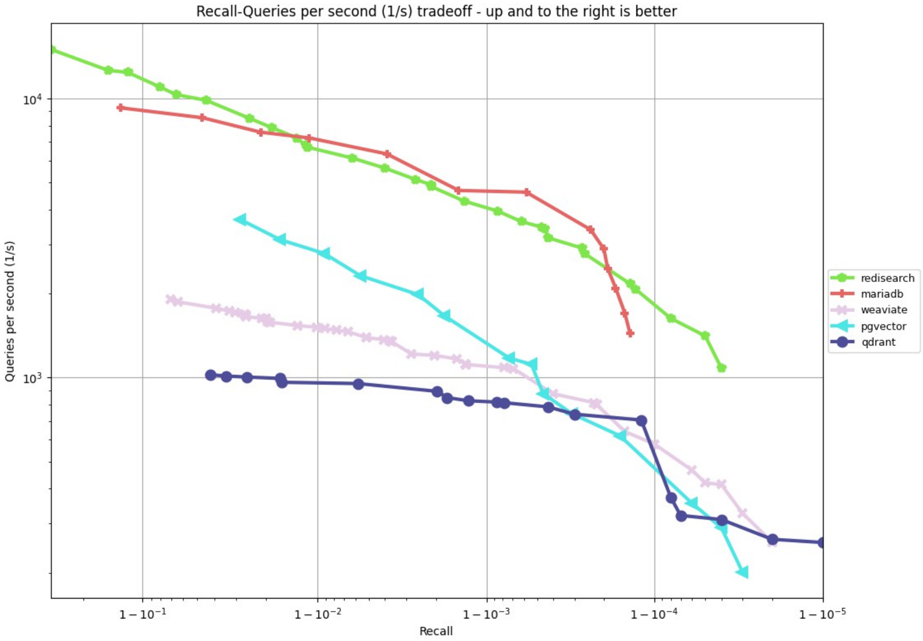
fashion-mnist-256-euclidean



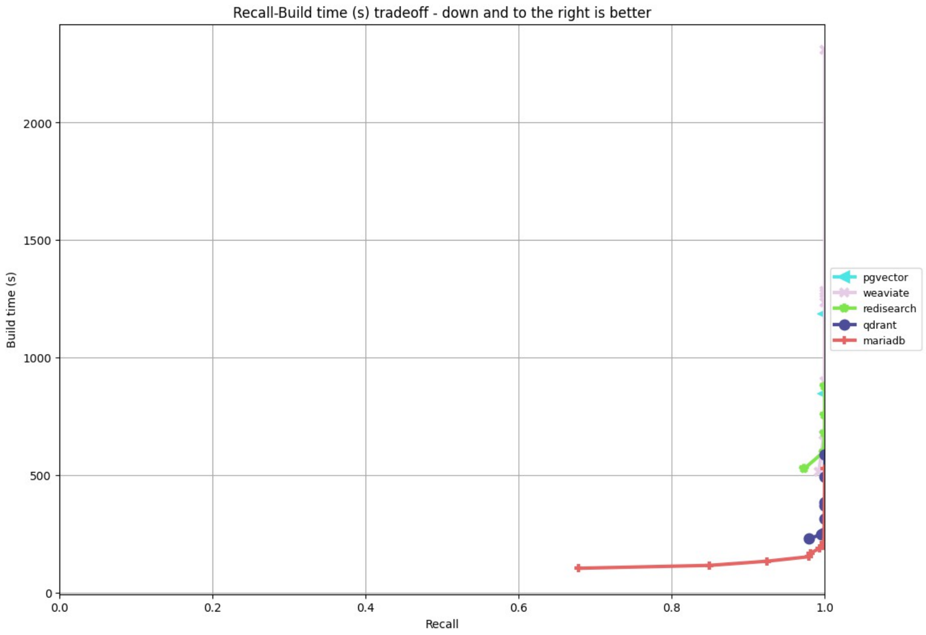


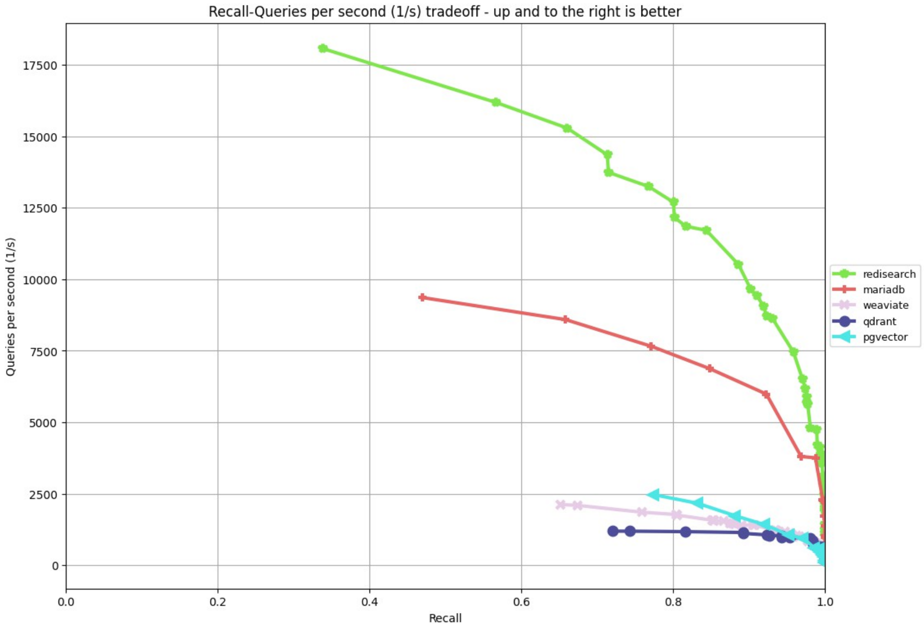
fashion-mnist-784-euclidean



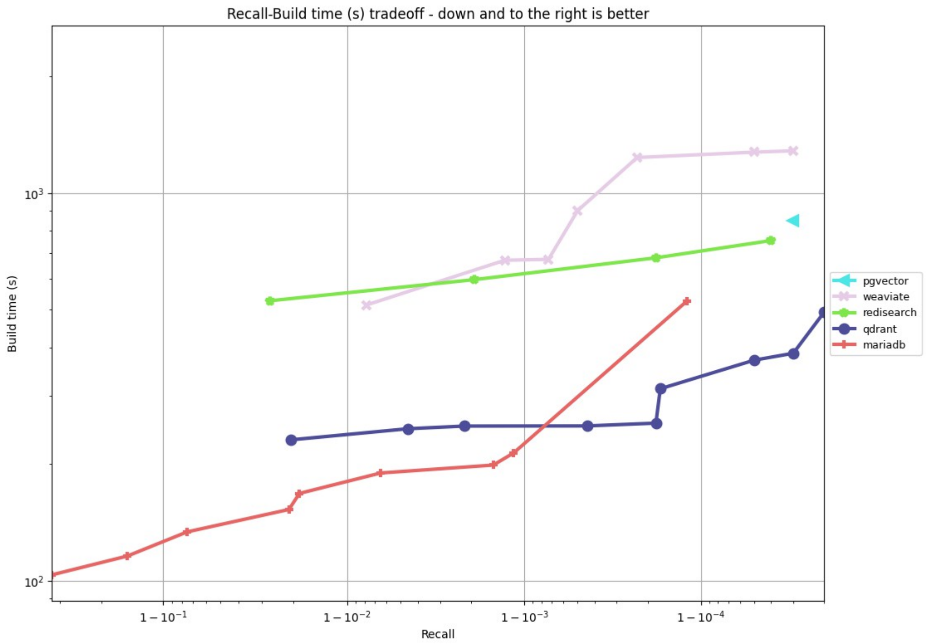


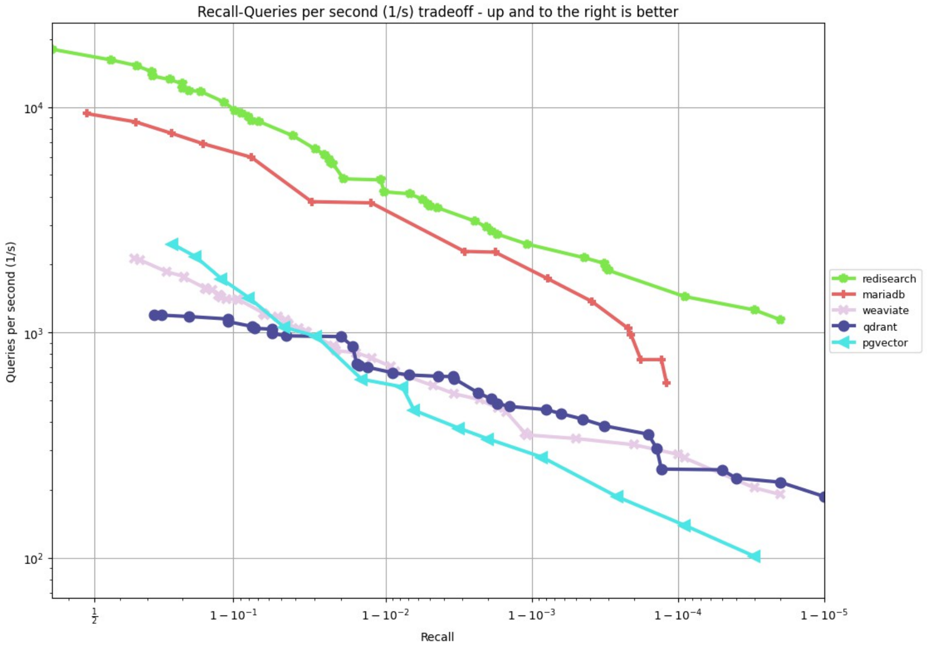
sift-128-euclidean


Recall-Build time (s) tradeoff - down and to the right is better

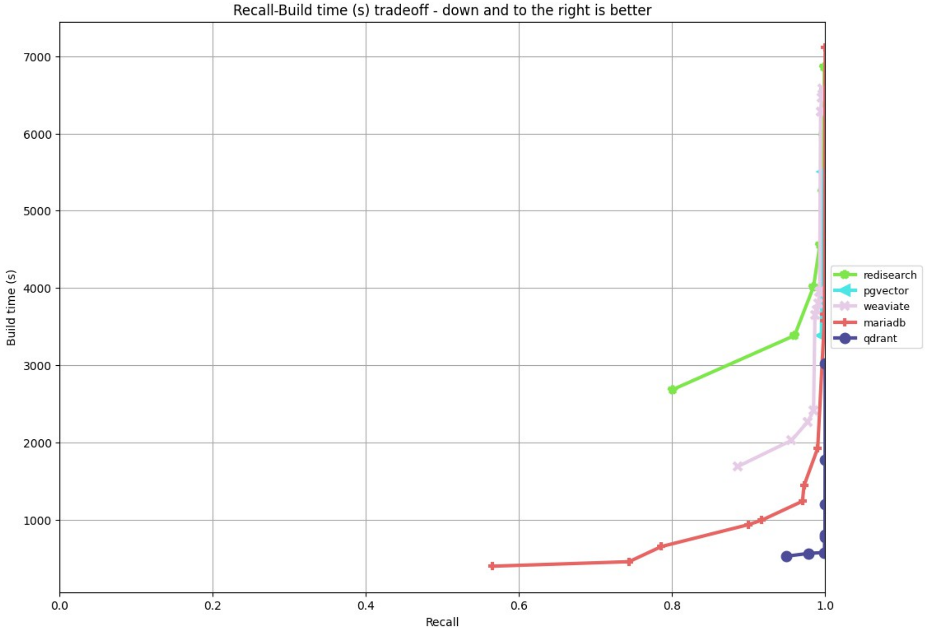

Recall-Queries per second (1/s) tradeoff - up and to the right is better

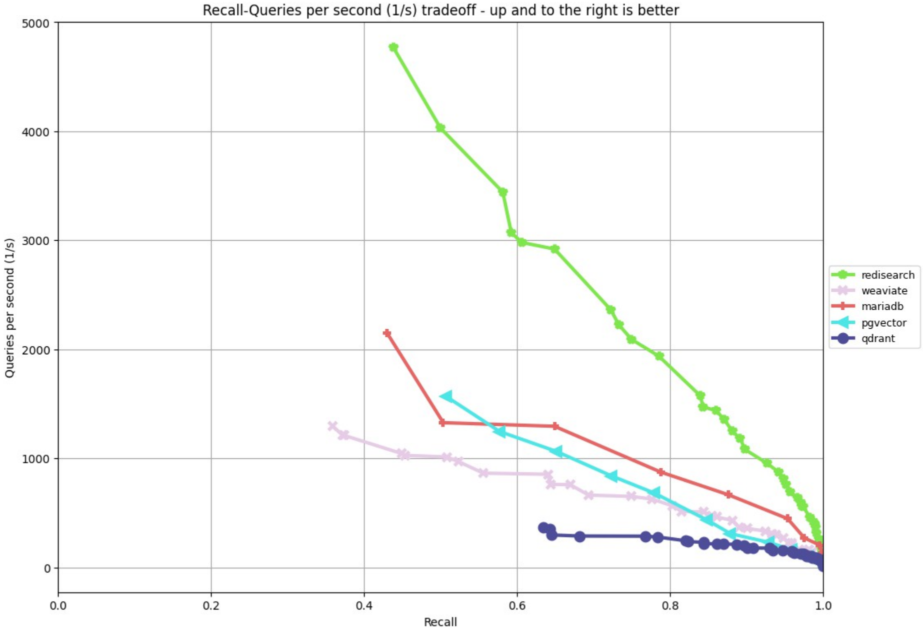
sift-128-euclidean



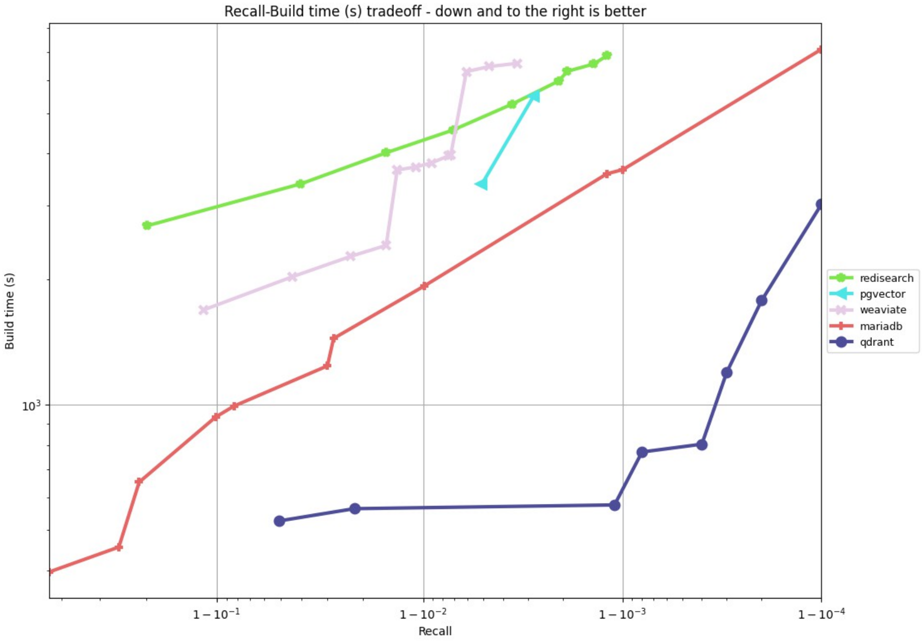


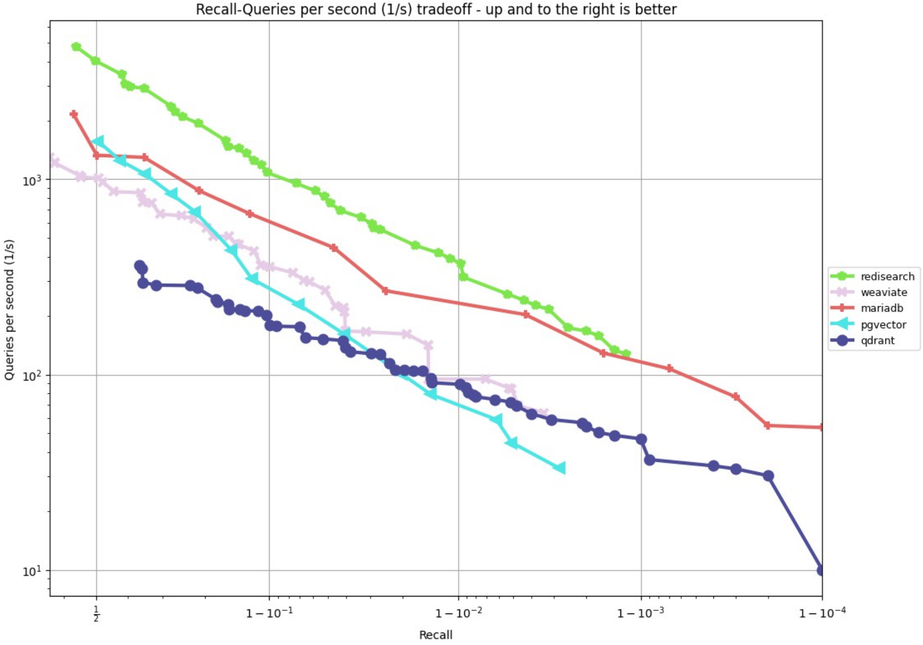
gist-960-euclidean





gist-960-euclidean





Server variables

- `mhnsw_cache_size`
- `mhnsw_distance_function`

- **`mhnsw_max_edges_per_node`** **(M)**
- **`mhnsw_min_limit`** **(ef)**

Where

- MariaDB Server 11.7 Preview
- MariaDB Server 11.7.1
  - likely, but not guaranteed

Future

Server Development

- Convenience:
  - `VECTOR(N)` data type
  - Observability
- Performance
  - Filtered Vector Search (with `WHERE` clause)
  - ARM64 optimizations
- More: MDEV-32887

Compatibility

- Langchain
- LlamaIndex
- ...?

Thank you!