1. Qdrant — распределённая и гибкая векторная база данных

- Тип: Open Source
- Алгоритмы поиска: HNSW (собственная реализация)
- Фильтрация по метаданным: Гибкая, поддерживает вложенную структуру и сложные иерархии
- Масштабирование: Да, поддерживается шардинг и репликация
- Квантование: Да, продуктовое и скалярное
- Use-case: Production-ready решения, работа с большими объёмами данных
- Сложность развёртывания: Средняя (требуется настройка кластеров и шардинга)

Описание:
Qdrant написан на Rust, подходит для продакшн-сценариев и больших объёмов данных. Поддерживает полноценную фильтрацию по метаданным, включая вложенные структуры и сложные фильтры.

---

2. Пример 1: NimbleNote — поиск заметок

- Исходные данные:
- 1-10 млн эмбеддингов
- Низкий QPS (<30)
- Простой стек, 1 сервер
- Важна быстрая MVP, минимальный DevOps
- Выбор: Chroma
- Лёгкая установка, не требует сложной настройки
- Новые условия:
- Гибкая фильтрация по метаданным (сложные AND/OR, range-фильтры)
- Необходимость горизонтального масштабирования
- Переход на: Qdrant

---

3. Пример 3: Каталог FinCommerce — поиск по товарам

- Характеристики:
- Production-система
- Сложные фильтры по метаданным (категории, диапазоны цен, теги, ACL)
- Десятки миллионов эмбеддингов
- Тысячи запросов в минуту
- Стабильная, но не гипермасштабируемая нагрузка
- Выбор: Qdrant
- Поддержка сложных фильтров
- Горизонтальное масштабирование коллекций
- Новые условия:
- Нагрузка выросла до сотен миллионов эмбеддингов, десятков тысяч QPS
- Необходимость независимого масштабирования поиска и записи
- Переход на: Milvus
- Микросервисная архитектура
- Возможность масштабирования search-write нод отдельно

---

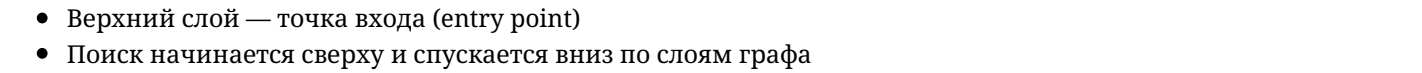
4. Алгоритм HNSW (Hierarchical Navigable Small World)

- Сложность поиска: O(log n)
- Структура: Многослойный граф
- Добавление элементов: Не требует полной перестройки структуры
- Фильтрация: Не поддерживает полноценную префильтрацию, только filtered search
- Оптимальность: Для одиночных запросов, не batch-поиска

Ограничения:

- Не поддерживает Product Quantization
- Не поддерживает шардинг на уровне алгоритма

Иллюстрация структуры HNSW:



- Верхний слой — точка входа (entry point)
- Поиск начинается сверху и спускается вниз по слоям графа
- Каждый слой содержит связанный граф узлов, где нижний слой — самый детальный

---

5. Дополнительные пояснения

- Qdrant поддерживает полноценную фильтрацию по метаданным с вложенной структурой и сложными иерархиями, в отличие от Chroma, где фильтрация базовая.
- Chroma не поддерживает горизонтальное масштабирование, поэтому Qdrant рассматривается как следующий шаг для production-сценариев.
- Milvus используется для очень больших нагрузок с микросервисной архитектурой и независимым масштабированием операций поиска и записи.