

УДК 004.65

**ВЕКТОРНЫЕ БАЗЫ ДАННЫХ: АРХИТЕКТУРА, ОСОБЕННОСТИ,
ПРЕИМУЩЕСТВА****Сагидова Марина Леонидовна,**кандидат технических наук, доцент, филиал ФГАОУ ВО «Мурманский арктический
университет» в г. Апатиты

Россия, Апатиты

sagidovaml@arcticsu.ru

Аннотация

В статье рассматривается архитектура и ключевые особенности векторных баз данных как стратегического компонента AI-систем. Анализируются их роль в обеспечении семантического поиска для задач машинного обучения, обработки естественного языка и компьютерного зрения. Описывается модульная структура, включающая специализированные слои хранения, ANN-индексы (HNSW, IVF) для быстрого поиска и подсистему метаданных. Выделяются преимущества — высокая производительность и масштабируемость, а также ограничения, такие как высокие требования к памяти. Подчеркивается, что векторные СУБД являются новым этапом в развитии систем хранения, необходимым для масштабируемого поиска по смыслу.

Ключевые слова: векторные базы данных, эмбединг, поиск сходства, многомерная индексация, HNSW, семантический поиск, ANN.

VECTOR DATABASES: ARCHITECTURE, FEATURES, ADVANTAGES**Sagidova Marina Leonidovna,**

Candidate of Technical Sciences, Associate Professor

Branch of Murmansk Arctic University in Apatity

Russia, Apatity

sagidovaml@arcticsu.ru

ABSTRACT

The article discusses the architecture and key features of vector databases as a strategic component of AI systems. Their role in providing semantic search for machine learning tasks, natural language processing, and computer vision is analyzed. A modular structure that includes specialized storage layers, ANN indexes (HNSW, IVF) for fast searching, and metadata subsystem is described. Advantages such as high performance and scalability are highlighted, along with limitations like high memory requirements. It is emphasized that vector DBMS represent a new stage in the development of data storage systems necessary for scalable meaning-based search.

Keywords: vector databases, embedding, similarity search, multidimensional indexing, HNSW, semantic search, ANN

Векторные системы управления базами данных (СУБД) стали ключевым элементом в разработке интеллектуальных приложений, что обусловлено стремительным развитием искусственного интеллекта (ИИ). Рост ИИ-индустрии повысил интерес к этим технологиям из-за необходимости эффективно работать с большими объёмами разнородных данных. Цель данной статьи — рассмотреть архитектуру и свойства векторных БД, продемонстрировать их роль как стратегически важного компонента современных AI-систем, а также проанализировать механизмы работы с данными, особенности поиска по сходству и преимущества таких решений.

Векторная база данных — это специализированная система хранения и поиска, предназначенная для работы с многомерными векторами (эмбедингами).

Традиционные СУБД (реляционные, документоориентированные, графовые) исторически проектировались для операций поиска по точным критериям: равенство, диапазон значений или сопоставление с шаблоном. Однако современные задачи машинного обучения (ML), обработки естественного языка (NLP) и компьютерного зрения (CV) требуют принципиально иного подхода — оценки семантического или функционального сходства между объектами [1]. Для решения этой задачи данные преобразуются в плотные числовые векторы фиксированной размерности. Схожие объекты располагаются в этом пространстве близко друг к другу. Таким образом, поиск сводится к нахождению ближайших соседей, что требует применения специализированных алгоритмов Approximate Nearest Neighbor (ANN), так как полный перебор (brute-force) неэффективен при работе с миллионами объектов.

Архитектура векторной базы данных

Архитектура любой векторной БД строится на модульном принципе для обеспечения эффективного хранения, индексации и поиска эмбедингов [2].

Хранилище векторов. Специализированный слой для хранения числовых представлений (эмбедингов) фиксированной размерности. Такой подход гарантирует компактность и оптимизированный доступ к данным.

Векторы хранятся в виде плоских файлов на диске. Для экономии места и ускорения ввода-вывода активно применяются методы сжатия, такие как Product Quantization (PQ). Данные часто сегментируются: новые векторы добавляются в новые сегменты, а старые оптимизируются. Для масштабирования используется шардинг (деление набора векторов на части, размещаемые на разных узлах кластера). Ключевое отличие от реляционных БД — гибкая схема метаданных, где каждый вектор может иметь произвольный набор атрибутов (ID, теги, ссылка на источник).

Индексный слой. Это ключевое звено архитектуры, реализующее ускоренный поиск по принципу сходства. Он использует структуры данных и алгоритмы (HNSW, IVF), а также библиотеки (FAISS). Благодаря индексам поиск ближайших соседей выполняется за миллисекунды даже в огромных массивах данных [3].

Векторные базы данных не используют полный перебор (brute-force), так как его вычислительная сложность ($O(Nd)$, где N — число векторов, а d — их размерность) делает поиск неприемлемо медленным при работе с миллионами объектов. Вместо этого применяются приближённые индексы ближайших соседей (ANN), которые находят не абсолютно точный, а очень близкий к оптимальному результат за короткое время.

Архитектурно слой индексации полностью отделён от слоя хранения данных. Это позволяет перестраивать или оптимизировать индексы с другими параметрами (например,

изменяя количество соседей в HNSW) без необходимости перезаписывать или перемещать сами векторы.

Наиболее распространённые алгоритмы ANN:

HNSW (Hierarchical Navigable Small World) — иерархическая графовая структура. Пространство векторов делится на слои: на верхних находятся «магистральные» узлы для быстрой навигации, а на нижних — плотные локальные кластеры. Поисковый запрос спускается по графу от грубых приближений к точным регионам. Этот алгоритм обеспечивает очень высокую скорость поиска (логарифмическая сложность), но требует значительного объёма оперативной памяти для хранения графа.

IVF (Inverted File Index) — пространство векторов предварительно разбивается на области (ячейки) с помощью алгоритмов кластеризации, таких как k-means. При поиске сначала определяются центроиды ячеек, ближайших к запросу, а затем просматриваются только векторы внутри этих ячеек. Это позволяет резко сократить количество вычислений.

PQ (Product Quantization) — метод сжатия векторов. Вектор разбивается на несколько подвекторов, для каждого из которых в кодовой книге подбирается ближайший по значению код. В итоге весь вектор заменяется набором компактных индексов. Это позволяет хранить и обрабатывать в оперативной памяти значительно большие коллекции данных, чем при использовании несжатых векторов.

Слой метаданных и документов. Обеспечивает связь между векторами и исходными объектами. Здесь хранятся ссылки на тексты, изображения, теги, метки и другую сопутствующую информацию, что позволяет не только находить похожие эмбединги, но и получать полные сведения об объектах.

Такая архитектура позволяет сочетать высокую скорость поиска с гибкой работой с семантическими связями и атрибутами данных.

Особенности векторных баз данных.

Приближённость и допустимая неточность. Векторные базы данных не гарантируют стопроцентной точности поиска, как это делают классические SQL-базы. Они находят наиболее похожие векторы, но иногда могут пропустить часть релевантных результатов. Для большинства задач машинного обучения такая небольшая потеря точности считается приемлемой [4].

Встроенная работа с моделями эмбедингов. База данных может напрямую использовать модели для преобразования текста, изображений или других данных в числовые векторы — например, через ONNX или SentenceTransformers. Это позволяет автоматизировать весь процесс: от получения исходных данных до их поиска [5].

Постепенное обновление индексов. При добавлении новых данных индекс для быстрого поиска не пересчитывается полностью, что экономит ресурсы. Вместо этого он обновляется по частям, а полная перестройка происходит позже, в фоновом режиме. На время обновления качество поиска может немного снизиться.

Гибкость в выборе метрик сравнения. Система поддерживает разные способы измерения «близости» векторов: косинусное расстояние, евклидово расстояние, скалярное произведение и другие. Это важно, так как для разных задач и моделей оптимальны разные метрики.

Версионирование моделей и миграция данных. Если меняется модель, которая создаёт векторы, старые и новые данные становятся несовместимыми — они «живут» в разных математических пространствах. Поэтому база данных должна уметь хранить разные версии векторов и помогать переносить данные при обновлении моделей.

Ключевые преимущества векторных баз данных

Высокая производительность. Векторные БД обеспечивают быстрый поиск даже среди миллиардов объектов. Благодаря специализированным индексам время поиска

растёт логарифмически с увеличением объёма данных, что позволяет получать результаты практически мгновенно [6].

Горизонтальная масштабируемость. Система легко расширяется за счёт добавления новых узлов в кластер. Данные автоматически перераспределяются (решардируются), что позволяет поддерживать высокую производительность и отказоустойчивость при росте нагрузки.

Согласованность данных. Векторы и связанные с ними метаданные обновляются атомарно. Это гарантирует, что между эмбедингом и его описанием не возникнет рассинхронизации, а данные всегда остаются целостными.

Удобство для разработчиков. Современные векторные БД предлагают высокоуровневые API, которые по синтаксису и логике напоминают SQL или запросы к документным хранилищам. Это упрощает интеграцию семантического поиска в приложения и снижает порог входа для разработчиков [6].

Ограничения векторных баз данных

Высокие требования к памяти. Индексы, особенно графы HNSW, могут занимать в 10 и более раз больше места, чем сами векторы. Это связано со сложной структурой связей между элементами для быстрого поиска, что увеличивает накладные расходы на хранение.

Ограниченная поддержка транзакций. В погоне за скоростью многие векторные СУБД жертвуют строгими ACID-гарантиями. Это затрудняет реализацию сложных транзакционных сценариев, где важна атомарность и согласованность операций.

«Холодный старт» индекса. Первоначальное построение индекса (например, для первого миллиона векторов) происходит медленно и требует значительных вычислительных ресурсов. Последующие обновления обычно выполняются быстрее, но начальная загрузка данных остаётся трудоёмкой задачей.

Векторные базы данных — новый этап в развитии систем хранения, который стал необходим из-за развития искусственного интеллекта. Благодаря своей архитектуре (специальные ANN-индексы, распределённые сегменты, смешанный поиск) они позволяют решать задачи, с которыми не справляются обычные базы данных. Главное их достоинство — возможность масштабируемого поиска по смыслу вместе с управлением метаданными. В будущем ожидается появление единых стандартов для работы с такими базами (аналогов SQL для векторов), развитие технологий постепенного обновления индексов и более тесная интеграция с потоковой обработкой данных.

Список литературы:

1. Зеляпугин И.А. Сравнение реляционных и векторных баз данных: проблемы и решения для обработки современных данных / И. А. Зеляпугин, М. А. Сафронов // Информационные технологии в науке и образовании. Проблемы и перспективы. Сборник статей по материалам XII Всероссийской научно-практической конференции. Пенза, 2025. С. 138-140.
2. Аверьянова А. Н. Проектирование баз данных для векторного поиска: как архитектура данных определяет эффективность современных AI-приложений / А. Н. Аверьянова, А. С. Демин // Цифровые, компьютерные и информационные технологии в науке и образовании. Сборник статей III Межрегиональной научно-практической конференции с международным участием. Брянск, 2026. С. 316-318.
3. Григорьев Ю.А. Анализ методов индексации в векторных базах данных // Информатика и системы управления. 2025. № 3 (85). С. 3-16.

4. Аменицкий А.В. Кибербезопасность. Лучшие векторные базы данных для раскрытия истинного потенциала ИИ // Актуальные вопросы науки, технологий и общества. Аленичева Т.С., Аменицкий А.В., Аменицкий Д.А., Аршун В.В., Атьман В.В., Балтабеков С.Б., Веремеев Р.Д., Зинюков Ю.В., Зонненберг Ю.Е., Колосова Е.Г., Кочергин И.Г., Куницын В.И., Лаврикова Н.И., Мамаев О.А., Мамаева Н.А., Расторгуева К.М., Рудикова-Фронхёфер Л.В., Рухович И.В., Сидорова И.Г., Фролов С.В. и др. Монография. Пенза, 2025. С. 121-133.
5. Коновалов Г. Г. Принципы архитектуры и преимущества векторных баз данных // Тенденции развития науки и образования. 2024. № 113-6. С. 57-59.
6. Сальников М. С. Векторные системы управления базами данных: основные принципы и преимущества / М. С. Сальников, А. А. Суханкин, А. С. Царев // Современные проблемы общества, науки и образования: от теории к практике. Сборник статей Международной научно-практической конференции. Пенза, 2024. С. 13-15.

References:

1. Zelyapugin, I.A. Comparison of Relational and Vector Databases: Problems and Solutions for Processing Modern Data / I.A. Zelyapugin, M.A. Safronov // Information Technologies in Science and Education. Problems and Prospects. Collection of articles based on the materials of the XII All-Russian Scientific and Practical Conference. Penza, 2025. pp. 138-140.
2. Averianova, A.N. Database Design for Vector Search: How Data Architecture Determines the Efficiency of Modern AI Applications / A.N. Averianova, A.S. Demin // Digital, Computer, and Information Technologies in Science and Education. Collection of articles from the III Interregional Scientific and Practical Conference with International Participation. Bryansk, 2026. pp. 316-318.
3. Grigoriev, Yu.A. Analysis of Indexing Methods in Vector Databases // Computer Science and Control Systems. 2025. No. 3 (85). pp. 3-16.
4. Amenitsky A.V. Cybersecurity. The best vector databases for unlocking the true potential of AI // Current issues in science, technology and society. Alenicheva T.S., Amenitsky A.V., Amenitsky D.A., Arshun V.V., Atman V.V., Baltabekov S.B., Veremeev R.D., Zinyukov Yu.V., Sonnenberg Yu.E., Kolosova E.G., Kochergin I.G., Kunitsyn V.I., Lavrikova N.I., Mamaev O.A., Mamaeva N.A., Rastorgueva K.M., Rudikova-Fronhoefer L.V., Rukhovich I.V., Sidorova I.G., Frolov S.V. and others. Monograph. Penza, 2025. pp. 121-133.
5. Konovalov G. G. Principles of Architecture and Advantages of Vector Databases // Trends in the Development of Science and Education. 2024. No. 113-6. pp. 57-59.
6. Salnikov M. S. Vector Database Management Systems: Basic Principles and Advantages / M. S. Salnikov, A. A. Sukhankin, A. S. Tsarev // Modern Problems of Society, Science, and Education: From Theory to Practice. Collection of Articles from the International Scientific and Practical Conference. Penza, 2024. pp. 13-15.