Vespa.ai

We Make AI Work

Built for Change: Why Vespa Outpaces Elasticsearch in Modern AI and Search Workloads

Architectural Differences Between Elasticsearch and Vespa

Vespa.ai

We are a platform for building and running real-time AI-driven applications for search, recommendation, personalization, and RAG. It enables enterprise-wide

AI deployment by efficiently managing data, inference, and logic, handling large

Contents

Introduction

Choosing the right search engine technology can transform your business operations by improving efficiency, user satisfaction, and scalability. This ebook dives into the architectural differences between Elasticsearch and Vespa, focusing on how each handles indexing, query processing, and scaling. By understanding these differences, you'll gain the clarity needed to choose a platform that supports your specific goals, whether it's enabling real-time updates, handling complex queries, or optimizing for batch processing.

The Write Path

Data Structures

Both Elasticsearch and Vespa use several core data structures to power their search capabilities:

- **Inverted Indices:** Enable fast full-text search by mapping unique tokens to the documents (and position) in which they appear. Both Elasticsearch and Vespa rely heavily on this structure for efficient lookup of text data.
- **HNSW Indexes:** To find neighbors to a query vector in vector and tensor data (ANN).
- **B-trees and dictionaries:** For fast searches in structured data
- **Columnar Stores:** Used for tasks like sorting, faceting, and ranking, ensuring fast retrieval and manipulation of field values across documents.
- **Raw Document Stores:** Retain original data for use in reindexing/redistribution, and retrieving document content.

The Write Path (cont.)

The main differences with respect to writes is in how the data structures are implemented:

Elasticsearch (Immutable segments)	Vespa (Mutable fields)
Data structures are built together for a given segment, and then remain unchanged when used for serving. Writes are handled by building new segments for a collection of incoming documents, and then searching in multiple such segments.	Uses mutable data structures per single field, which are updated in place. This allows real-time writes, and sidesteps the need for searching multiple segments.

Availability of New Documents

Most applications constantly ingest additional data, and in most being able to serve fresh data leads to higher perceived quality.

Elasticsearch (Refresh Required)	Vespa (Available Immediately
Because Elasticsearch is based on Lucene immutable index segments, new data doesn't become searchable until the next refresh operation, which occurs asynchronously. While this allows Elasticsearch to process writes in batches, it introduces a delay that impacts real-time applications.	Vespa uses native real-time updatable index structures, ensuring that data is searchable immediately, and before each write is acknowledged to the user. This means updates are immediately visible to users, and enables building interactive workflows where the next view depends on a write being observable.

Handling Updates to Documents

Modern applications will typically need to update certain fields of documents across an existing corpus with high frequency. Examples include making budget updates in ad serving system on every ad impression, and updating behavioral signals such as clicks in real time.

Elasticsearch (Full document writes)	Vespa (Individual in-place field updates)
Any change to a document field requires the entire document to be read and then reindexed into a new segment. This means that updates even to simple field values are as expensive as writing and indexing the entire document anew.	With Vespa's in-place field-level writers, changes to individual fields can be made individually, ensuring that they are applied in real time and that writes to simple fields are cheaper and faster..

The Query Path

The query path determines how effectively a search engine delivers results, directly influencing customer satisfaction, operational efficiency, and overall success. It also determines a system's ability to scale and maintain performance under heavy demand, such as during sales events or viral content surges.

Businesses have unique search requirements, ranging from real-time updates and hybrid search capabilities to complex filtering. A flexible query path enables the system to adapt to these needs, supporting advanced features like personalized recommendations, semantic search, and dynamic filtering. This section explores the key differences between Elasticsearch and Vespa in threading models, query handling, and filtering techniques, highlighting their implications for your business.

Threading Models

Modern data server nodes consist of a large number of cores that can execute operations in parallel (multithreading). To allow a high query throughput and utilize all these cores, both Elasticsearch and Vespa use many threads to handle different queries at the same time. However, to deliver low latency, it is important to also be able to use multiple cores in parallel to process a single query.

Elasticsearch is only able to use a single thread per query when searching a segment, which means that more cores can't help in reducing latency. Vespa offers a fully flexible model where any number of cores can be used to process the same query in parallel, which ensures that low latency can be delivered when processing capacity is available, regardless of the amount of data and the complexity of queries.

Mutable vs. Immutable Data Structures: Impact on queries

With Elasticsearch immutable segments, each query must in practice run the search on multiple segments on each node, containing old and newer data - about four segments is common in steady state. This increases the cost of each query in all cases, but only moderately with text indexes since segments containing newer data are small and cost is proportional to size. The big difference is seen with modern vector (HNSW) indexes. Since the cost of searching such indexes are largely independent of size, the cost of searching four segments is about 4x that of searching a single index.

As Vespa uses mutable data structures and avoids segments, it avoids this cost multiplier. This effect in combination with Vespa's higher native performance is how the engine becomes more than 10x more efficient than Elasticsearch on vector and hybrid searches.

Scalability

Both systems scale horizontally but differ in their underlying models. Elasticsearch scales by spreading indices and shards across nodes, making data movement efficient. However, this approach has tradeoffs involving uneven load distribution, hot spots, and potential bottlenecks which must be made by application developers and which are hard to change once made. Performance test findings show that Vespa consistently outperforms Elasticsearch across hybrid, lexical, and vector search workloads in both throughput and latency. Refer to section 8.2.2 of the full performance report for detailed findings.

Vespa automatically distributes data in much more granular buckets. This granular distribution allows for more efficient resource utilization, without burdening developers with making hard choices, and scales to any amount of data. As data can be redistributed automatically between nodes on the granular bucket level, Vespa supports scaling clusters both up and down in size, which simplifies operations and allows for cost optimization over time.

Practical Considerations for Business Leaders

Selecting the right platform depends on aligning its capabilities with your business goals and operational priorities. Each excels in different areas, making one a better fit depending on your specific challenges.

Elasticsearch is a strong choice for applications that focus on batch processing and historical data analysis with few queries where low latency and high availability is less important than low cost. It's well-suited for handling large volumes of immutable data, such as log analysis, archival searches, and reporting dashboards. Its architecture prioritizes efficient write-once performance, making it a cost-effective solution for workloads where query speed is less critical.

Vespa, on the other hand, is built for real-time applications where immediate data visibility and query efficiency are essential. This applies to both end-user experiences and internal processes, ensuring seamless updates to existing data. Vespa excels in use cases like deep research and agentic RAG applications, recommendation systems, dynamic marketplaces, and personalized content platforms—scenarios that demand fast, adaptable, and scalable search capabilities. Additionally, it is well-suited for advanced document search and voice-based chat applications, enabling cutting-edge real-time interactions.

With its ability to efficiently handle frequent updates and deliver superior query performance, Vespa also optimizes infrastructure costs. Its speed not only enhances user experience but also reduces the total cost of ownership, particularly in GPU-intensive tasks such as vector search. By scaling seamlessly across massive clusters, Vespa ensures high-performance, cost-efficient search and AI-driven applications in real time.

If real-time data visibility, high-speed queries, and frequent updates are your priorities, Vespa is the better fit. If your focus is on batch processing, historical analysis, or cost-effective storage of immutable data, Elasticsearch may be the right choice.

For a deeper comparison of performance, see the Elasticsearch and Vespa benchmarking report. By understanding each platform's strengths, you can confidently choose the best tool to power your search and data processing needs.

Next Steps

Now that you understand the key differences between Elasticsearch and Vespa, the next step is to evaluate which platform best meets your needs. Choosing the right solution requires a clear understanding of your workload, business priorities, and performance expectations.

Here's how you can move forward:

- Assess your workload's indexing and query requirements
- Identify key business metrics like real-time updates, query latency, or batch efficiency
- Test both platforms with sample workloads to evaluate performance
- Refer to technical benchmarks and tuning guides for deeper insights
- Read the Elasticsearch vs. Vespa performance report
- Read the Vinted's technical blog about their journey from Elasticsearch to Vespa

By aligning your platform choice with your operational goals, you'll set your business up for continued success.

Vinted's Switch from Elasticsearch to Vespa

"We chose Vespa because it's a modern, reliable and performant search engine that enables advanced features like image search, recommendation, ML ranking, and more. Our choice to go with Vespa is already paying off, as our members are able to find what they need more often, which leads to an increased number of purchases and higher total merchandising value. And looking to the future, we see that Vespa will continue to be a key building block and will enable new and impactful experiences for our members." *Mindaugas Mozūras, CTO, Vinted*

At a Glance

- 21 European markets plus USA.
- 120 million registered users.
- 77 million monthly visitors.
- 1 billion active searchable items.
- 25,000 item searches per second.
- 10,300 requests per second data update/remove operations.

Introduction

Vinted.com, an online platform for buying and selling second-hand items, has built its success around a big idea: make second-hand a first choice. The platform enables users to easily list, discover, and purchase pre-owned clothing, electronics and home decoration, catering to a community-driven marketplace where buyers and sellers interact directly. Robust recommendation, search, and personalization engines deliver an exemplary online experience, maximize customer engagement and grow loyalty:

Advanced search functionality allows users to find items that fit specific needs and preferences from a staggering 1 billion searchable items for sale.

Recommendation engines suggest items based on individual browsing and purchasing habits, increasing the likelihood of conversions and enhancing user satisfaction. Personalization shapes each user's journey according to their style and shopping behaviors. With tailored suggestions, from trending brands to favored categories, Vinted effectively keeps users engaged and simplifies the discovery of relevant products.

Vinted's environmentally conscious alternative to traditional retail and superior online customer experience has led to impressive revenue growth and strategic expansion. In 2023, Vinted's revenue jumped by 61% year-over-year, reaching approximately €596 million. The platform now has a user base exceeding 120 million and offers one billion items for sale, establishing itself as a dominant player in the online resale market.

High-Performance To Enable Growth

Vinted Engineering faced a challenge as rapid growth strained their existing Elasticsearch implementation for recommendation, search, and personalization, leading to high operational costs and the need for hardware upgrades. To address this, they sought a more efficient solution and, after a thorough evaluation, chose Vespa in 2023 for its scalability, high performance, automated management, and ability to support both vector and traditional search in one system.

High performance in Vespa is achieved with its distributed and balanced architecture, which ensures scalability and fault tolerance by distributing data, queries, and machine learning models across multiple nodes. This allows it to scale horizontally and vertically, increasing capacity and performance. By performing computations where data is stored, Vespa reduces data transfer costs and latency, enhancing the online experience by providing visitors with faster, high-quality recommendations and boosting engagement and revenue. Vespa's highly efficient architecture also immediately impacted running costs by reducing the number of servers from 120 to 60. Search result consistency has also improved as all traffic is managed within a single Vespa cluster, reducing search latency by 2.5 times.

Vespa's balanced distributed architecture evenly spreads the workload across all nodes, preventing "hot node" bottlenecks. It automatically redistributes content when content groups are changed or added, reducing the need for performance tuning and making it easy for Vinted to optimize resources. This ensures consistent performance, even when traffic patterns fluctuate.

The improved performance efficiency has also allowed a more than threefold increase in ranking depth, enabling the algorithm to consider up to 200,000 candidate items when determining which products to display in search results.

Greater Search Accuracy

Search accuracy directly impacts user satisfaction, engagement, and, ultimately, Vinted's success. Users who receive highly relevant results are more likely to engage with the platform, make purchases, or return for future visits, driving higher conversion rates and retention.

Vinted found querying in Vespa to offer significant advantages, presenting a marked shift from their previous experience with Elasticsearch. Vespa achieves search accuracy through its hybrid search and advanced ranking capabilities. It supports vector and traditional text-based search and uses machine learning models to rank content effectively. This means Vespa can understand and match complex user queries more intelligently, considering the meaning of words, user preferences, and the search context, and do so in real-time.

With over a billion constantly changing items in its inventory, efficiently organizing and preparing data for fast and accurate retrieval is essential. This process, known as indexing, transforms unstructured information, such as text from documents or data streams, into a structured format, making it easier to deliver relevant search results and power applications like recommendation systems and AI-driven search.

This ensures that the right information can be accessed instantly, supporting business decisions and enhancing customer experiences. With Vespa's real-time indexing capability—unavailable in Elasticsearch—the delay for changes to appear in search results dropped from 300 seconds, greatly enhancing responsiveness and the user experience.

Business Impact

Vinted has commended Vespa for its pragmatic approach to problem-solving and the team's genuine commitment to support. The Vespa team's active engagement has been instrumental in enhancing Vinted's customer engagement strategy, leading to notable business outcomes. Improved search capabilities with Vespa delivered measurable results: transactions purchased rose by 1.1%, and Gross Merchandise Value (GMV) increased by 0.6%. This uplift equates to over €3.5 million in additional merchandise moving through the platform, driven by Vespa's search enhancements. Furthermore, the Total Cost of Ownership of the Vespa implementation was approximately half that of the previous system, offering a highly cost-effective solution that simultaneously boosted user engagement and conversions. This combination of reduced costs and improved search performance has directly contributed to revenue growth and enhanced customer satisfaction by making relevant products easier to find.

Vespa.ai is a platform for building and running real-time AI-driven applications for search, recommendation, personalization, and RAG. It enables enterprise-wide AI deployment by efficiently managing data, inference, and logic, handling large data volumes and over 100K queries per second. Vespa supports precise hybrid search across vectors, text, and structured metadata. Available as both a managed service and open source, it's trusted by organizations like Spotify, Vinted, Wix, and Yahoo. The platform offers robust APIs, SDKs for integration, comprehensive monitoring metrics, and customizable features for optimized performance.

Interested to learn more? We have many different resources and information available through our social platforms

GitHub Twitter LinkedIn YouTube