

**Executive summary**

# Elasticsearch VS Vespa **Performance Comparison**

The Evolving Demands of Search



# Executive Summary: Elasticsearch vs Vespa Performance Comparison

## Introduction: The Evolving Demands of Search

Modern applications face relentless pressure to deliver real-time updates, manage increasing traffic, and control operational costs while delivering an exceptional customer experience. For systems with high query rates or requiring frequent updates to data or signals, selecting the right search technology directly impacts performance, scalability, and cost efficiency. To empower enterprise leaders with actionable, data-driven insights, Vespa conducted a comprehensive benchmarking analysis, comparing Vespa and Elasticsearch under real-world scenarios. This report focuses on key metrics such as write performance, query efficiency, and scalability. By leveraging these insights, organizations across industries can make informed decisions to enhance user experience, optimize resource allocation, and drive innovation.

## Benchmarking Focus: Metrics That Matter for Search Success

The evaluation focused on key performance metrics that directly impact operational costs, customer satisfaction, and business scalability:

- **Write Performance:** How quickly and efficiently can the system handle new documents, product updates, pricing changes, and inventory adjustments?
- **Query Performance:** How well does the system process diverse search queries, ensuring fast and relevant results?
- **Scalability:** Can the platform manage increasing demand without performance degradation during peak periods like holiday sales or flash promotions?
- **Cost Efficiency:** Does the platform minimize infrastructure requirements and expenses while maintaining high performance?

This executive summary highlights each platform's strengths and trade-offs for businesses aiming to enhance their search capabilities and maintain a competitive edge. Explore the [full report](#) for in-depth data and technical insights to guide your decision-making.

## Performance: A Key Driver for Success

Efficiently managing write and query workloads is critical for search-driven systems, as these directly influence customer experience, operational efficiency, and the ability to respond to

rapidly changing market demands. To thrive in a competitive environment, platforms must balance speed, scalability, and resource efficiency.

## Write Performance: Keeping Your Data Up-to-Date

Efficient write performance ensures that new documents, product updates, pricing changes, and inventory adjustments are reflected quickly in search results. The benchmark report evaluates two critical metrics:

- **Throughput (Operations/Second):** Measures how quickly the system can handle write operations.
- **Throughput Per CPU Core:** Evaluates resource efficiency, showing how much work the system accomplishes per unit of processing power.

### Key Findings

The findings reveal that Vespa excels in real-time updates with high throughput, enabling quick changes that enhance customer satisfaction. Its efficient use of CPU resources reduces infrastructure costs while maintaining performance, making it ideal for dynamic, fast-paced platforms. While Elasticsearch is well-suited for bulk data ingestion and static scenarios, it is less effective for real-time operations.

The results of write performance are summarized below. Refer to **section 8.1** of the [full report](#) for detailed findings.

#### **Throughput (Operations/Second):**

- Elasticsearch processes operations more quickly during the initial population of the search index (from 0 to a full dataset). This is beneficial when building a search system from scratch or ingesting large datasets in bulk.
- Vespa delivers 9x higher throughput for updates, such as price changes or behavior signals like click counts, making it more efficient for environments requiring frequent, real-time modifications. This is critical for modern applications where the accuracy and speed of updates directly impact customer experience.
- Vespa outperforms Elasticsearch in 3 out of 4 workloads, including reindexing, reindexing while running queries, and update. However, neither engine utilized 100% of CPU capacity, indicating potential bottlenecks in threading or contention points.

#### **Throughput Per CPU Core:**

- Elasticsearch demonstrates higher efficiency in resource utilization during reindexing and reindexing while running queries workloads, achieving more with less CPU utilization.
- Vespa performs better in the refeed\_with\_queries workload, likely due to more efficient query handling, as the measured CPU usage includes queries.
- Vespa outperforms Elasticsearch in the update workload, achieving 4x better throughput per CPU core.

For applications requiring frequent updates and dynamic search capabilities, Vespa's ability to handle real-time updates at scale ensures customer satisfaction and operational agility.

## Query Performance: Delivering Fast and Relevant Results

Search query performance refers to how efficiently the system processes user searches. Modern applications require fast and scalable search systems that can handle complex queries with minimal latency for a better user experience. This involves two critical metrics:

- **Throughput:** The number of queries a system can handle per second. Higher throughput allows the platform to serve more users simultaneously without degrading performance.
- **Latency:** The time it takes to process a single query. Lower latency ensures users receive fast, responsive search results, enhancing the overall experience.

Faster query responses reduce user frustration and enhance satisfaction. Low latency is especially critical during high-demand periods like sales events, where delays can lead to lost revenue. Higher throughput ensures the search system can accommodate large volumes of concurrent users, essential for scaling as the business grows. This is particularly important during peak traffic events.

The following three types of queries were benchmarked for this report, which represent the types of searches performed in most use cases:

- **Lexical search:** Traditional keyword search over the title and description fields.
- **Vector search:** Using text embeddings powered by approximate nearest neighbor search over the dense vector representations of the products.
- **Hybrid search:** Combination of the above query types.

## Key Findings

The findings reveal that Vespa significantly outperforms Elasticsearch in both throughput and latency. The results of query performance are summarized below. Refer to **section 8.2** of the [full report](#) for detailed findings.

- **Hybrid Searches:** Vespa is 4x more efficient, crucial for complex multi-criteria searches.
- **Vector Searches:** Vespa achieves 6x better throughput, ideal for advanced recommendation systems.
- **Lexical Searches:** Vespa's 2.7x throughput advantage improves traditional keyword-based queries.

Vespa also provides consistently 2x–6x lower latency, ensuring fast, responsive results even under heavy traffic.

Vespa's superior performance in vector and hybrid searches can enhance product recommendations and result relevance, boosting conversion rates. Its efficient handling of queries with fewer resources reduces infrastructure costs while maintaining high performance. This empowers businesses to exceed user expectations, fostering loyalty and repeat purchases.

## Scalability: Meeting Growing Demands with Efficiency

The ability to scale efficiently under high traffic is essential for modern applications, especially during peak demand periods like holiday sales or flash promotions. This analysis identifies which system can manage increasing query loads without compromising performance, helping businesses plan for growth and avoid disruptions. The full report evaluates how Vespa and Elasticsearch scale under varying query loads and concurrency levels, focusing on hybrid, lexical, and vector searches. Key metrics include average latency, 99th percentile (99p) latency, and CPU usage, providing insights into each system's efficiency and limitations.

### Key Findings

The 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 report](#) for detailed findings.

- **Hybrid search:** Vespa handles up to 5,000 QPS with 64 concurrent clients while maintaining low latency and CPU usage.
- **Lexical search:** Vespa manages 9,500 QPS with stable latency, significantly outperforming Elasticsearch.
- **Vector search:** Vespa delivers 10,000 QPS with latency under 5ms, even under high concurrency.

For businesses relying on hybrid search capabilities to deliver personalized and dynamic user experiences, Vespa's scalability, low latency, and resource efficiency make it the better choice. Elasticsearch's limitations, particularly in the multi-segment configuration, restrict its ability to handle high-concurrency hybrid searches effectively, making it less suitable for modern, high-demand environments.

## Cost Efficiency: Achieving More with Less

Cost efficiency is not just about saving money—it's about creating a sustainable, scalable, and competitive business. For industries with tight margins and high customer expectations, optimizing costs ensures profitability, enables growth, and enhances the ability to deliver exceptional value to customers.

### Key Findings

For a system handling 700 queries per second (QPS), 300 document writes per second, and maintaining sub-50ms query times, Vespa's efficient resource utilization results in 5x lower infrastructure costs compared to Elasticsearch. Refer to **section 10** of the [full report](#) for detailed findings.

- **Elasticsearch:** Needs 41 CPU cores to meet SLA requirements, requiring a 48-core c8g.12xl instance costing \$1.90/hour (AWS EC2).
- **Vespa:** Meets the same SLA with only 5.1 CPU cores, using an 8-core c8g.2xl instance or m8g.2xl instance at \$0.36/hour.

Vespa's superior throughput and lower latency with fewer resources significantly reduce infrastructure costs. Vespa's cost efficiency is more than just a technical advantage—it provides businesses with a strategic edge. It enables operational scaling and new functionality: you can have fewer clusters, or clusters with fewer nodes, which are less troublesome to maintain. You can also support more complex queries or more queries per user request, delivering more relevant results to clients. By lowering infrastructure costs, enabling scalability, and allowing better resource allocation, Vespa helps businesses optimize profitability, stay competitive, and invest in growth and innovation.

## Conclusion: Empowering Applications for the Future

For applications with high query and write demands, Vespa offers substantial cost and efficiency advantages over Elasticsearch. It delivers faster query responses, better resource utilization, and significant savings in cloud infrastructure costs, making it the optimal choice for businesses prioritizing performance and scalability.

Vespa excels in real-time indexing, high-concurrency handling, and query performance, making it ideal for applications requiring frequent updates or high query rates. Its ability to process more queries per second with significantly lower latency—while using fewer resources—translates into substantial cost savings and greater operational efficiency. This positions Vespa as a strategic asset for businesses looking to optimize profitability and scalability.

Conversely, Elasticsearch demonstrates strengths in initial indexing with minimal updates, making it better suited for use cases where data is only written once and never updated, and where query rates are low.

By choosing Vespa, organizations across industries can achieve lower infrastructure costs, better resource allocation, and enhanced user experiences by keeping data up-to-date and including real-time signals. Vespa supports seamless operations and empowers businesses to invest in growth, innovation, and competitive differentiation in an increasingly demanding marketplace.

Explore the [full report](#) for detailed performance data, architectural insights, and cost simulations.