[Nagarro]-[Ecommerce]

DAR Document

Nagarro Software Pvt. Ltd.

Akshat Aggarwal

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| Version | Date | Author/Contributor | Comments |
| 1 | January 28, 2024 | Akshat Aggarwal | Initial Draft |
| 2 | February 08, 2025 | Akshat Aggarwal | Updated Comparison Analysis matrix |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1 Introduction 4](#_Toc421883713)

[1.1 Objective and scope of document 4](#_Toc421883714)

[2 Requirements at a Glance 5](#_Toc421883715)

[3 Available tools 6](#_Toc421883716)

[3.1 Mongo DB 6](#_Toc421883717)

[3.1.1 Features 6](#_Toc421883718)

[3.1.2 Pricing 6](#_Toc421883719)

[3.2 Cassandra DB 6](#_Toc421883720)

[3.2.1 Features 6](#_Toc421883721)

[3.2.2 Pricing 6](#_Toc421883722)

[4 Comparison Analysis 7](#_Toc421883723)

[4.1 Point Matrix 7](#_Toc421883724)

[4.2 Comparison 1: Performance, Speed and Availability 7](#_Toc421883725)

[4.3 Comparison 2: Scalability, Flexibility and Ease of Use 7](#_Toc421883726)

[5 Recommendation 8](#_Toc421883727)

[6 Assumptions 9](#_Toc421883728)

[7 Risks 10](#_Toc421883729)

[8 Appendix 11](#_Toc421883730)

[8.1 References 11](#_Toc421883731)

# Introduction

This document provides a comparative analysis between MongoDB and Cassandra to determine the optimal database solution for storing product data and cart data in an e-commerce application. Both databases offer NoSQL storage capabilities but vary in terms of performance, scalability, and flexibility, which are essential for enhancing the application’s performance.

## Objective and scope of document

The objective of this document is to evaluate the suitability of MongoDB and Cassandra for storing product data and cart data in the e-commerce platform. This analysis will focus on key aspects like performance, scalability, and data handling capabilities, providing a recommendation on the best fit for ensuring optimal application performance.

# Requirements at a Glance

* **High Performance for Read and Write Operations:** The database should support low latency read and write operations.
* **Scalability:** Ability to scale horizontally to accommodate growing product data and user traffic.
* **Support for Unstructured Data:** Flexible schema design to manage diverse product attributes and descriptions.
* **Fault Tolerance and Availability:** Ensuring data availability with minimal downtime.
* **Efficient Data Replication and Consistency Options:** Support for flexible consistency models and effective replication mechanisms.

# Available tools

* Mongo DB
* Cassandra DB

## Mongo DB

MongoDB is a document-oriented NoSQL database that uses BSON (binary JSON) to store data, which makes it suitable for applications requiring flexible schemas and support for unstructured data.

### Features

* Document-based storage, allowing flexible schema for semi-structured data.
* Powerful indexing for fast query performance.
* Horizontal scaling via sharding.
* Replica sets for high availability and fault tolerance.
* Advanced querying, aggregation, and analytics capabilities.
* Strong community support and managed services through MongoDB Atlas.
* Delivers solid read and write performance, especially with indexing.

### Pricing

Free and open source, with paid enterprise versions and managed services available on MongoDB Atlas.

## Cassandra

Cassandra is a distributed NoSQL database designed for high availability and fault tolerance. It is optimized for handling large-scale, write-intensive applications.

### Features

* Wide-column store with a flexible data model for high write throughput.
* Peer-to-peer architecture, with no single point of failure.
* Horizontal scalability with easy replication across multiple data centers.
* Tunable consistency levels for flexible read and write consistency.
* Optimized for high availability and fault tolerance, making it suitable for mission-critical applications.

### Pricing

Free and open source, with enterprise support and managed services available from DataStax.

# Comparison Analysis

This section compares MongoDB and Cassandra based on essential criteria for storing product data in an e-commerce environment.

## Weightage Matrix

|  |  |
| --- | --- |
| Feature | Points |
| Performance (Read & Write) | 5 |
| Speed | 5 |
| Scalability | 5 |
| Flexibility | 4 |
| Availability & Fault Tolerance | 4 |
| Ease of Use | 3 |

## Comparison 1: Performance (Write and Read), Speed and Availability

|  |  |  |
| --- | --- | --- |
| Feature | Mongo DB | Cassandra DB |
| Read Performance | **5** | **4** |
| Write Performance | **4** | **5** |
| Speed | **5** | **4** |
| Availability & Fault Tolerance | **4** | **5** |

## Comparison 2: Scalability, Flexibility and Ease of Use

|  |  |  |
| --- | --- | --- |
| Feature | Mongo DB | Cassandra DB |
| Scalability | **4** | **5** |
| Flexibility | **5** | **3** |
| Ease of Use | **5** | **3** |

# Recommendation

MongoDB is recommended for the e-commerce application’s product and cart database. Its document-based model offers high flexibility, making it well-suited for handling diverse product attributes, descriptions, and unstructured data. MongoDB’s powerful indexing, advanced querying capabilities, and ease of use make it an excellent choice for an e-commerce platform with dynamic product data and variable user queries. While Cassandra is advantageous for write-heavy and globally distributed use cases, MongoDB’s schema flexibility and rich querying capabilities make it a better fit for product data storage in this e-commerce application.

# Assumptions

1. The application needs low-latency read operations to support fast product searches and an enhanced user experience.
2. Flexibility in schema design is essential to accommodate different product attributes and descriptions.
3. High availability and minimal downtime are key requirements for maintaining user experience.
4. Data consistency requirements are moderate, allowing MongoDB’s eventual consistency to support most operations effectively.

# Risks

* MongoDB: Potential performance degradation in high-write environments if sharding is not configured optimally.
* Cassandra: Limited flexibility for complex queries and indexing, which may require additional data modeling effort if chosen over MongoDB.

# Appendix

## References

1. https://www.mongodb.com/
2. https://cassandra.apache.org/\_/index.html
3. https://www.mongodb.com/resources/compare/cassandra-vs-mongodb