A Polyglot Data Architecture for Scalable E-Commerce Analytics

Student Name: **Gabin MBISHINZEMUNGU** Student ID: **101090** Course: Big

February 2, 2026

Abstract

E-commerce platforms operate at the intersection of transactional workloads, behavioral event streams, and analytical reporting. Managing these heterogeneous data patterns using a single storage or processing technology often results in performance bottlenecks and limited scalability. This report presents an alternative implementation of a polyglot big data analytics system that combines MongoDB, Apache HBase, and Apache Spark to address distinct data access and analysis requirements. MongoDB is used to manage core business entities and support aggregation-based analytics, HBase is employed for large-scale session and activity logs, and Spark performs distributed batch analytics across datasets. Through revenue analysis, customer behavior exploration, and integrated cross-system queries, the study demonstrates how a multi-model architecture improves analytical flexibility and supports data-driven decision-making in an e-commerce context.

1 Introduction

Digital commerce ecosystems continuously generate large volumes of diverse data, ranging from structured transactional records to high-frequency interaction logs. These datasets vary not only in size but also in access patterns, update frequency, and analytical value. As data volumes grow, traditional monolithic database solutions struggle to efficiently serve operational queries while simultaneously supporting large-scale analytics.

This project adopts a polyglot persistence strategy, selecting specialized technologies based on the nature of the data and expected query workloads. By distributing responsibilities across MongoDB, HBase, and Spark, the system achieves both scalability and analytical depth.

2 Problem Definition

Online retail systems must simultaneously support:

- Flexible storage of evolving business data such as products, customers, and orders.
- High-throughput ingestion of time-ordered browsing and session events.
- Computationally intensive analytics for reporting, optimization, and strategic insights.

Attempting to satisfy all these requirements with a single database technology introduces inefficiencies. This project explores how a multi-model approach mitigates these limitations.

3 Project Objectives

3.1 Primary Objective

To design and evaluate a scalable big data analytics architecture for e-commerce using multiple complementary data technologies.

3.2 Specific Objectives

- Model and store transactional and reference data using a document-oriented database.
- Persist high-volume session logs in a wide-column store optimized for sequential access.
- Execute distributed batch analytics using Apache Spark.
- Perform integrated analytics that combine behavioral and transactional perspectives.
- Derive interpretable business insights from analytical results.

4 Application Context

In a real-world e-commerce setting, stakeholders require insights into customer purchasing trends, popular product categories, and the relationship between user engagement and revenue generation. Data-driven answers to these questions enable inventory planning, personalized marketing, and customer retention strategies. The architecture implemented in this project mirrors these operational realities using synthetic but realistic data.

5 Dataset Overview

The dataset used in this study was synthetically generated using Python utilities and consists of multiple JSON files representing distinct business entities and activity logs. Key components include:

- User profiles with geographic and registration metadata.
- Product and category information with pricing and inventory attributes.
- Transaction records containing nested purchase items.
- Browsing session logs capturing page views, device context, and session duration.

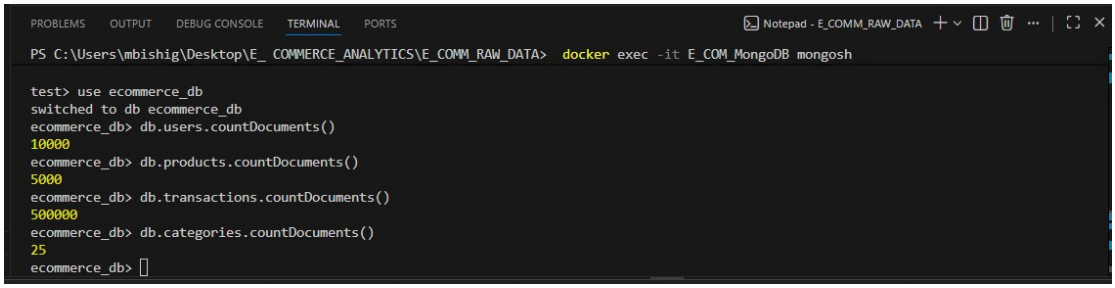

Figure 1: Verifying the count of dataset entities and relationships (users, sessions, transactions, products, categories)..

6 System Architecture

6.1 Architectural Overview

The system architecture is composed of three primary layers: storage, processing, and analytics. Each layer leverages a technology optimized for its role.

6.2 Document Store: MongoDB

MongoDB is responsible for storing semi-structured business data such as products, users, and transactions. Its flexible schema design supports nested documents, making it suitable for transaction records with variable line items. Aggregation pipelines enable efficient computation of metrics such as revenue by category and customer spending summaries.

6.3 Wide-Column Store: HBase

HBase is used to manage large-scale browsing session data. The wide-column model supports sparse attributes and efficient time-based access patterns. By designing row keys that combine user identifiers and timestamps, the system enables fast retrieval of user activity histories.

6.4 Distributed Processing: Apache Spark

Apache Spark provides the computational backbone for batch analytics. By loading JSON data into distributed DataFrames, Spark performs large-scale aggregations, joins, and transformations that would be impractical on a single machine.

7 Implementation Strategy

7.1 MongoDB Data Modeling and Analytics

Collections were created for users, products, categories, and transactions. Analytical queries were implemented using aggregation pipelines to compute metrics such as total revenue per category and top-selling products.

7.2 HBase Session Storage

Session data was ingested into HBase using a streaming approach to handle large file sizes efficiently. Column families were defined to logically separate metadata, device information, and event payloads.

7.3 Spark Batch Analytics

Spark batch jobs processed transaction and session data to generate analytical outputs including customer rankings and product affinity indicators. Spark SQL was used to express complex queries in a declarative manner.

8 Integrated Analytics

A cross-system analytical task was implemented to examine the relationship between user engagement and spending behavior. Engagement metrics derived from HBase session logs were combined with transaction aggregates from MongoDB.

9 Results and Discussion

The analysis revealed a positive association between frequent platform usage and higher spending, although variability exists across users. Some highly engaged users exhibit low spending behavior, suggesting opportunities for targeted conversion strategies.

10 Business Insights

Key insights derived from the analytics include:

- Identification of high-revenue product categories for strategic focus.
- Recognition of high-value customers suitable for loyalty programs.
- Detection of product affinity patterns that can inform recommendation systems.

11 Scalability and Limitations

While the architecture demonstrates scalability through distributed storage and processing, limitations include the use of synthetic data and simplified deployment configurations. Future implementations could integrate real-time streaming and cloud-based infrastructure.

12 Conclusion

This project validates the effectiveness of a polyglot big data architecture for e-commerce analytics. By aligning data technologies with workload characteristics, the system achieves improved performance, scalability, and analytical richness. The approach provides a practical blueprint for modern data-driven e-commerce platforms.

13 Future Work

Potential extensions include real-time stream processing, advanced recommendation algorithms, and enhanced customer segmentation models.

References

[1] MongoDB Documentation: Data Modeling and Aggregation Framework.

[2] Apache HBase Reference Guide.

[3] Apache Spark Documentation.

[4] Faker Python Library Documentation.