A COMPARATIVE STUDY OF LEADING NO SQL TECHNOLOGIES FOR DATA WAREHOUSING: UNVEILING PERFORMANCE

Introduction

In the era of big data, data warehousing transitions from relational to diverse NoSQL technologies. With numerous NoSQL options available, selecting the optimal database is paramount for high performance and scalability. This poster delves into the exploration of two seminal papers that dissect the performance intricacies of NoSQL databases with aim to provide valuable insights for organizations seeking optimal solutions for their data warehousing needs.

Critical Review

From the two seminal papers, the first paper (Kausar, et al., 2022) evaluated the performance of three prominent No SQL databases MongoDB, Cassandra, and Redis using YCSB method. The authors used various overall throughput levels (100,000 to 1,000,000 operations) to compare performance. Additionally, authors measured average latency in diverse workload scenarios encompassing a mix of read, write, and update activities.

The findings of the paper are:

- Redis stands out in overall throughput and read operations in specific scenarios.
- Cassandra exhibits greater insert latency but demonstrates stability in certain workloads.
- MongoDB excels across diverse workloads.

Workload	Database	Throughput	Read Latency	Update/Insert Latency
A (50% Read and 50% Update)	MongoDB	High	Lower	Lower
A (50% Read and 50% Update)	Redis	High	Moderate	Moderate
A (50% Read and 50% Update)	Cassandra	Moderate	High	Stable
B (95% Read and 5% Update)	MongoDB	Higher	Lower	Moderate
B (95% Read and 5% Update)	Redis	Moderate	Lower	Moderate
B (95% Read and 5% Update)	Cassandra	Moderate	High	Stable
C (100% Read)	MongoDB	High	Lower	N/A
C (100% Read)	Redis	Moderate	Lower	N/A
C (100% Read)	Cassandra	Moderate	High	N/A
D (5% Insert and 95% Read)	MongoDB	Higher	Lower	Moderate
D (5% Insert and 95% Read)	Redis	Higher	Lower	Moderate
D (5% Insert and 95% Read)	Cassandra	Moderate	Higher	Stable
E (95% Scan and 5% Insert)	MongoDB	Higher	N/A	Lower
E (95% Scan and 5% Insert)	Redis	Moderate	N/A	Higher
E (95% Scan and 5% Insert)	Cassandra	Moderate	N/A	Moderate

Table 1: Detailed Comparison Table

The second paper (Carvalho, et al., 2023) evaluated the performance of three document-based No SQL databases —MongoDB, CouchDB, and Couchbase— using YCSB method.

The authors used the following workloads for performance evaluation.

Workloads	Operations	Distribution	Records	Threads	Data Size
A—Update Heavy	Read: 50% Update: 50%	Zipfian	100,000 1,000,000 10,000,000	1 3 6	Field size = 500 bytes Field number = 20 500 bytes × 20 = 10 KB
B—Read Mostly	Read: 95% Update: 5%	Zipfian	100,000 1,000,000 10,000,000	1 3 6	Field size = 500 bytes Field number = 20 500 bytes × 20 = 10 KB
C—Read Only	Read: 100%	Zipfian	100,000 1,000,000 10,000,000	1 3 6	Field size = 500 bytes Field number = 20 500 bytes × 20 = 10 KB
D—Read Latest	Read: 95% Insert: 5%	Latest	100,000 1,000,000 10,000,000	1 3 6	Field size = 500 bytes Field number = 20 500 bytes × 20 = 10 KB
E—Short Ranges	Scan: 95% Insert: 5%	Zipfian Uniform	100,000 1,000,000 10,000,000	1 3 6	Field size = 500 bytes Field number = 20 500 bytes × 20 = 10 KB
F—Read–Modify–Write	Read: 50% Read–Modify–Write: 50%	Zipfian	100,000 1,000,000 10,000,000	1 3 6	Field size = 500 bytes Field number = 20 500 bytes × 20 = 10 KB
G—Update Mostly	Update: 95% Read: 5%	Zipfian	100,000 1,000,000 10,000,000	1 3 6	Field size = 500 bytes Field number = 20 500 bytes × 20 = 10 KB
H—Update Only	Update: 100%	Zipfian	100,000 1,000,000 10,000,000	1 3 6	Field size = 500 bytes Field number = 20 500 bytes × 20 = 10 KB

Table 1: Workloads used to evaluate the NoSQL databases

Critical Review Contd...

The findings of paper are:

- CouchDB performs better than MongoDB and Couchbase in scale-up scenarios, excelling with varying thread counts.
- MongoDB excels in read-intensive workloads, showcasing its flexibility with dynamic schemas (except for scan operations).

Figure: Log runtime representation, in seconds, of all workloads except workload E

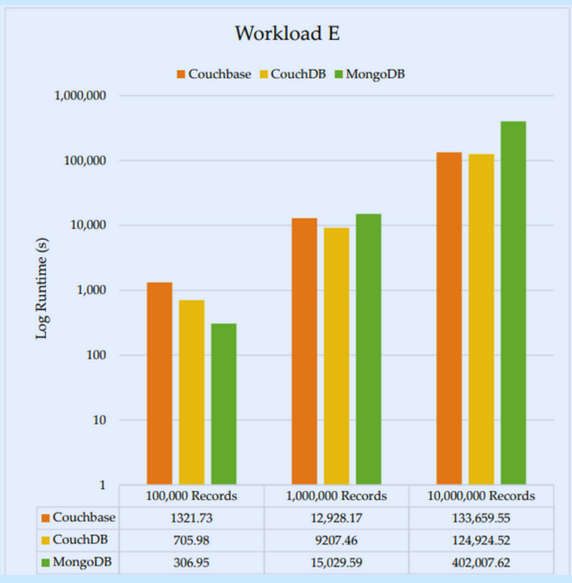

Figure: Log runtime representation, in seconds, of all workloads except workload E

MongoDB is consistently highlighted across both papers, making it a strong contender for heavy workload scenarios in modern data warehousing.

While other databases may not be suited for heavy workloads, they find application in different scenarios.

R**edis**: Exceptional overall throughput and read operation performance make it valuable for efficient data retrieval in data warehousing (Charan, et al., 2022).

Cassandra: Despite greater insert latency, is suitable for data warehousing prioritizing consistency and reliability in certain workloads (Wahid & Kashyap, 2019).

CouchDB: Valuable for data warehouses that require scalability and flexible thread management

Summary

In summary, the evaluation of two papers comparing NoSQL databases reveals MongoDB's consistent excellence, positioning it as a versatile choice for large and responsive data handling. Redis excels in rapid data retrieval, while Cassandra and CouchDB find niche applications in scalability and flexibility scenarios, respectively. The findings guide informed decisions for optimal NoSQL database selection in modern data warehousing.

References

- Carvalho, I., Sá , F. & Bernardino, J., 2023. Performance Evaluation of NoSQL Document Databases: Couchbase, CouchDB, and MongoDB. p. 17.
- Charan, P. S. B. et al., 2022. REDIS: IN MEMORY DATA STORE. Volume 12.
- Kausar, M. A., Nasar, M. & Soosaimanickam, A., 2022. A Study of Performance and Comparison of NoSQL Databases: MongoDB, Cassandra, and Redis Using YCSB. INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY, p. 10.
- Wahid, A. & Kashyap, K., 2019. Cassandra—A Distributed Database. p. 8.

Student Name: Prativa Chhantyal Student ID/Registration No: 239587769