**Performance Benchmarking of Key-Value Store NoSQL Databases**

**Advance Database Systems- Dr. Xiaofei Zhang**

Sasidhar Yemukapati U00865672

Divya Reddy Kankanala U00871092

Viswas Ayyappa Konda U00869343

# Introduction

In the rapidly evolving landscape of data management, the efficient handling of unstructured data has become paramount, particularly in the realms of mobile computing and social networking. Our project conducts a comparative analysis of Redis, MongoDB and SSDB—three prominent NoSQL databases within the key-value store landscape. Redis, known for its exceptional data access speed, is evaluated alongside SSDB, which offers a durable approach to data storage and MongoDB with its unique architecture and operational capabilities. This benchmarking project seeks to quantify their performance and uncover the underlying behaviors that drive their effectiveness in practical applications. The insights taken from this study aim to clarify the practical implications of utilizing these databases in key-value storage roles, a critical component in the infrastructure of modern applications.

The motivation behind this study stems from the escalating demand for robust and scalable data management systems capable of handling varied workloads with optimal efficiency. Redis, with its high-performance in-memory data store capabilities, SSDB a persistent storage system, and MongoDB, a document-oriented database, have been pivotal in this landscape. However, with their differing approaches to data management, a comprehensive understanding of their performance under varied conditions is crucial for informed decision-making in system architecture.

Our objective is to benchmark these databases using the Yahoo! Cloud Serving Benchmark (YCSB), assessing them across a spectrum of workloads (from Workload A to F). This benchmarking is designed to reveal insights into their relative advantages and drawbacks, especially in terms of throughput, latency, and scalability under different operational scenarios.

The challenges in this project are multifaceted. Firstly, ensuring consistent performance metrics in a single-node environment, which may not entirely reflect real-world, multi-node deployments, poses a significant constraint. Secondly, discerning the performance edge between the three databases, particularly when they exhibit convergent behaviors at lower thread counts and divergence at higher loads, adds complexity to the analysis. Lastly, the varied performance behavior in diverse workloads, ranging from heavy read to write operations, demands a nuanced examination.

This study, therefore, serves as a critical exploration of Redis, MongoDB and SSDB, aiming to provide clear insights into their operational efficacies and situational aptness, guiding users in the selection of the appropriate database technology based on their specific application requirements.

# Related Work

The Performance benchmarking in NoSQL databases has attracted considerable attention due to the critical role these systems play in managing big data. Studies targeting the evaluation of these databases often focus on scalability and performance in varied operational environments. [1] carried out a comparative analysis highlighting the capabilities of NoSQL databases like MongoDB and Cassandra within cloud-based architectures. Despite the depth of their research, SSDB was notably absent, leaving a gap in the collective understanding of persistent storage performance when compared to in-memory databases like Redis.

The work of [2] expanded the scope of such comparisons to include a broad spectrum of databases, analyzing key performance metrics such as throughput and latency [2]. This analysis was comprehensive, yet it stopped short of exploring the impact of different workload types as defined by the Yahoo! Cloud Serving Benchmark (YCSB), which are pivotal for appraising performance in practical settings.

[3] investigated various NoSQL databases, assessing their suitability for big data analytics. While their research was extensive, it lacked a head-to-head comparison between two of the significant players in the field: Redis and SSDB [3]. This omission is significant, as each database brings distinct advantages and limitations that are only revealed through direct comparison.

Our project distinguishes itself by zeroing in on a side-by-side performance analysis of Redis and SSDB, specifically employing the YCSB workloads to dissect database behavior in scenarios ranging from read-intensive operations to transactions involving complex write-read-modify cycles. We concentrate on single-node database instances, which mirrors the operational constraints typical in small to midsize enterprise settings [4]. This focus presents a unique contribution to the field, offering insights that are especially relevant to organizations operating within such infrastructural bounds.

This comparative study is not just about measuring performance; it's about contextualizing it within the framework of real-world application demands. By doing so, it equips stakeholders with the detailed information necessary to make well-informed decisions tailored to their specific application requirements [5].

# Solution Section

The core of our project lies in the methodical benchmarking of Redis, MongoDB and SSDB, utilizing the Yahoo! Cloud Serving Benchmark (YCSB) to facilitate a rigorous and standardized evaluation. The solution design consists of multiple distinct modules, each of which contributes to a comprehensive framework for analysis.

## Benchmarking Tool Selection

Choosing the Yahoo! Cloud Serving Benchmark (YCSB) for our database performance assessment was influenced by its well-established reputation and comprehensive testing capabilities. YCSB provides diverse workloads that simulate a broad spectrum of real-world operations, including those that are read-dominant and write-dominant, thus ensuring our evaluation reflects a wide array of practical scenarios [1]. This choice was vital in ensuring that our analysis would be both relevant and on par with standards recognized by the research community at large.

## Database Setup

We implemented single-node instances of Redis, MongoDB and SSDB. Redis was set up to function in its default in-memory mode, where data is stored and accessed directly from volatile memory, offering high throughput and low latency, while MongoDB was set up with the default configuration parameters and SSDB was configured with its standard persistent storage mechanism, providing a different approach to data management by ensuring data durability across sessions. The objective of this experimental configuration was to simulate normal usage scenarios, hence offering valuable insights into the performance of these databases in commonly encountered environments.

## Workload Customization and Execution

We tailored the Yahoo! Cloud Serving Benchmark (YCSB) workloads, identified as A through F, to assess distinct dimensions of database functionality. Workload A was configured to weigh heavily on update operations. In contrast, workloads B and C prioritized read operations, workload D targeted the retrieval of the most recent entries, workload E emphasized on scanning operations, and workload F involved sequences of reading, altering, and writing data. This tailored approach allowed us to conduct an equitable analysis of both databases under the same testing conditions.

## Data Collection and Analysis

Data was gathered pertaining to runtime, throughput, operations count, and average delay. The data was systematically evaluated in order to extract patterns and insights, with a specific emphasis on evaluating the performance of each database across various workload types.

## Performance Evaluation

Our evaluation rigorously compared the operational efficiency of Redis, MongoDB and SSDB, measuring response time, workload scalability, and operational throughput to determine their performance under diverse conditions.

Our solution was designed with a commitment to impartial and thorough benchmarking, targeting standardized workloads to capture extensive performance data. This strategy ensured a strong and unbiased comparison of Redis, MongoDB and SSDB, revealing their respective proficiencies and constraints, thus informing the optimal database choice for distinct user requirements.

# Evaluation Section

In assessing the performance of Redis, MongoDB and SSDB, we carefully maintained a controlled environment to ensure the integrity of the resultant data. We utilized a Kali Linux virtual environment on VMware Workstation 16 Pro, supported by a Windows 10 Pro host. The Kali system leveraged an Intel(R) Core(TM) i5-7200U CPU at 2.50GHz, supported by 8 GB of RAM on the host, mirroring a configuration often seen in small to medium-scale deployments, thus ensuring the applicability of our findings to common operational scenarios.

The benchmarks for our evaluation were obtained using the Yahoo! Cloud Serving Benchmark (YCSB), renowned for its effectiveness in testing database performance across a range of workloads that simulate real-world operations. The workloads ranged from A to F, encompassing scenarios from read-heavy to write-heavy operations. This breadth ensured a comprehensive assessment of each database's capabilities.

Our results indicated a stark contrast in performance between Redis, MongoDB and SSDB. Redis's in-memory data storage architecture yielded much lower runtimes across all workloads, particularly excelling in workload D, which involved read-modify-write operations. In contrast, MongoDB, with its disk-based storage mechanism, demonstrated higher throughput, particularly in workloads B through F, indicative of its superior performance in handling operations at scale In contrast, SSDB, with its disk-based storage mechanism, demonstrated higher latencies, indicative of the additional time required for disk I/O operations. Despite this, SSDB showed respectable throughput rates, particularly in workload C, suggesting its capability to handle operations effectively at scale. while MongoDB is favourable in scenarios where high throughput is required Notably, Redis showcased its performance prowess with significantly higher throughput and lower latencies in most workloads, underlining the advantages of in-memory processing (see Table 1).

**Table 1: Benchmarking Results Summary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **index** | **Database** | **Workload** | **RunTime(ms)** | **Throughput(ops/sec)** | **Operations** | **AverageLatency(us)** |
| 0 | Redis | A | 177071 | 564.745 | 100000 | 1755.87 |
| 1 | Redis | B | 688 | 1453.488 | 494 | 487.89 |
| 2 | Redis | C | 17567 | 5692.492 | 100000 | 169.34 |
| 3 | Redis | D | 513 | 1949.318 | 953 | 266.94 |
| 4 | Redis | E | 13410 | 7457.122 | 100000 | 128.07 |
| 5 | Redis | F | 485 | 2061.856 | 1000 | 271.06 |
| 6 | SSDB | A | 63579 | 1572.846 | 100000 | 628.592 |
| 7 | SSDB | B | 1069 | 935.454 | 517 | 1122.021 |
| 8 | SSDB | C | 55891 | 1789.197 | 100000 | 550.64 |
| 9 | SSDB | D | 1040 | 961.538 | 935 | 723.75 |
| 10 | SSDB | E | 58197 | 1718.302 | 100000 | 572.98 |
| 11 | SSDB | F | 902 | 1108.647 | 1000 | 655.404 |
| 12 | MongoDB | A | 177071 | 564.745 | 100000 | 1755.87 |
| 13 | MongoDB | B | 17567 | 5692.492 | 100000 | 169.34 |
| 14 | MongoDB | C | 13410 | 7457.122 | 100000 | 128.07 |
| 15 | MongoDB | D | 13696 | 7301.402 | 100000 | 130.99 |
| 16 | MongoDB | E | 12320 | 8116.883 | 100000 | 117.37 |
| 17 | MongoDB | F | 13283 | 7528.42 | 100000 | 127.2 |

The visual representation of our data further emphasizes these findings. Figure 2 illustrates the runtime comparison, showing Redis's faster processing times across all workloads, with a particularly significant difference in Workload A. Figure 1 compares throughput, where Redis generally outperforms SSDB, most notably in Workloads C and E, highlighting its rapid in-memory operations. Figure 1 presents the average latency, with Redis demonstrating lower values in every workload, which underscores its efficiency for in-memory data handling compared to SSDB's disk-based storage mechanism.

A comparison of different colored bars

Description automatically generated with medium confidence

*Fig 1. Performance Comparison: Throughput, and Average Latency*

From these observations, it is evident that the choice between Redis, MongoDB and SSDB should be influenced by the specific performance needs of the application. Redis is the preferred solution when low latency and fast data access are paramount, while MongoDB is favourable in scenarios where high throughput is required and SSDB could be considered in scenarios where the persistence of data and cost factors are more critical despite the trade-off in latency.

A graph with different colored bars

Description automatically generated

*Fig2. Performance Comparison: Runtime*

The experimental findings serve as a testament to the importance of selecting the right database based on the targeted workload characteristics. They underscore the trade-offs between in-memory and disk-based NoSQL databases, and the impact these have on the performance metrics crucial to the operation of data-driven applications.

## CONCLUSION

This study has critically assessed Redis, MongoDB and SSDB through YCSB workloads, revealing Redis's low-latency advantage, MongoDB's throughput efficiency for scalable operations and SSDB's operational effectiveness despite its higher latency due to disk-based storage. Future work may explore these databases in multi-node environments and consider the implications of data persistence strategies, aiming to optimize data management in complex applications.

# References

[1] A. Al-Sakran and M. J. Aqel, "A Comparative study of NoSQL Databases," in *Proc. International Conference on Communication, Management and Information Technology*, Apr. 2017.

[2] A. Gandini, M. Gribaudo, W. J. Knottenbelt, R. Osman, and P. Piazzolla, "Performance Evaluation of NoSQL Databases," in *A. Horváth and K. Wolter, Eds., Computer Performance Engineering, EPEW 2014, Lecture Notes in Computer Science, vol. 8721*, Springer, Cham, 2014, doi: 10.1007/978-3-319-10885-8\_2.

[3] A. B. M. Moniruzzaman and S. A. Hossain, "NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison," June 2013.

[4] W. Khan, T. Kumar, C. Zhang, K. Raj, A. M. Roy, and B. Luo, "SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review," Big Data Cogn. Comput., vol. 7, no. 97, 2023. [Online]. Available: <https://doi.org/10.3390/bdcc7020097>

[5] J. R. Lourenço, B. Cabral, P. Carreiro, et al., "Choosing the right NoSQL database for the job: A quality attribute evaluation," *J. Big Data*, vol. 2, no. 18, 2015. [Online]. Available: <https://doi.org/10.1186/s40537-015-0025-0>.

[6] B. F. Cooper, A. Silberstein, E. Tam, et al., "Benchmarking cloud serving systems with YCSB," in *Proc. 1st ACM Symp. on Cloud Computing, SoCC 2010*, Indianapolis, Indiana, USA, June 10-11, 2010, doi: 10.1145/1807128.1807152.