Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
18/2 (2025), 251-259. DOI: http://dx.doi.org/10.21609/jiki.v18i2.1530

Comparative Evaluation of Database Systems for High-Volume Seismic Prediction Data Management in Real-Time Applications

Ari Wibisono, Rafif Naufal Rahmadika

Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia

Email: ari.w@cs.ui.ac.id

Abstract

The Earthquake Early Warning System (EEWS) plays a pivotal role in mitigating structural damage and minimizing casualties by issuing alerts prior to the arrival of destructive seismic waves (S-waves), through the detection of the earlier and faster P-waves. The operational effectiveness of EEWS depends not only on the accuracy of its predictive algorithms but also on the efficiency of the underlying data storage and management infrastructure. This study presents a comparative evaluation of three data storage approaches—MongoDB, MongoDB with sharding, and InfluxDB—as well as the MiniSEED (mseed) binary format, with a focus on their performance in managing real-time seismic prediction data. Benchmarking was conducted based on two key metrics: Input/Output Operations Per Second (IOPS) and data throughput. The results indicate that both MongoDB and InfluxDB offer strong performance in high-ingestion scenarios, with MongoDB demonstrating higher IOPS, while InfluxDB exhibits better scalability and consistency as data volume increases. Conversely, the mseed format achieves exceptionally high throughput due to its flat-file structure but lacks the responsiveness and query capabilities required for real-time analytics. These findings suggest that MongoDB and InfluxDB are well-suited for integration into scalable EEWS infrastructures, offering a balance between performance and flexibility. Future work will extend this evaluation to larger-scale datasets and alternative architectures such as data lake systems to improve disaster response readiness.

Keywords— Earthquake early warning system, MongoDB, InfluxDB, database efficiency, IOPS, data throughput

1. Introduction

Originally proposed by Dr. Cooper in 1868, Earthquake Early Warning Systems (EEWS) represent a class of technologies intended to issue rapid alerts prior to the arrival of destructive secondary (S) waves, utilizing the faster-traveling primary (P) waves as early indicators [1]–[3]. Central to this mechanism is the prompt detection of P-waves, which provides a critical time window for initiating protective measures aimed at minimizing structural damage and human casualties before the arrival of more harmful seismic activity. Through EEWS, vital safety protocols—such as automatically unlocking emergency exits, decelerating or pausing escalators and elevators, and halting operations in sensitive environments like industrial plants and nuclear facilities—can be enacted [4].

Progress in the fields of seismology and artificial intelligence (AI) over recent decades has facilitated the creation of increasingly precise models for earthquake prediction. These models are capable of generating forecasts that encompass essential parameters, including the epicenter location, magnitude, timing, and projected impact. Nevertheless, the effectiveness of such predictive models is not exclusively determined by their computational accuracy; the ability to manage and store the resulting data efficiently is equally fundamental to their success.

A significant challenge associated with earthquake prediction systems pertains to ensuring both scalability and rapid data access. Given that earthquake forecasts are produced at high frequency and in substantial quantities, the underlying storage infrastructure must be capable of supporting continuous real-time data ingestion. Furthermore, the information must be structured in a manner that permits rapid retrieval by end-users, particularly in scenarios where time-sensitive decisions—such as initiating evacuations or triggering early warning mechanisms—are required.

Over the past several decades, seismic monitoring efforts have resulted in the accumulation

of vast quantities of waveform data. As of April 2022, for instance, the Incorporated Research Institutions for Seismology (IRIS) had amassed approximately 800 terabytes of seismic records [5]. In addition, considerably larger datasets are managed by various local and regional seismic networks. The emergence of distributed acoustic sensing (DAS) as a data collection methodology is anticipated to significantly increase the velocity and volume of seismic data acquisition [6]. Consequently, conventional data access paradigms—where users are required to download extensive datasets for local analysis—are increasingly being recognized as unsustainable.

The growing complexity of seismic sensor networks, along with advances in real-time analytics, has emphasized the need for adaptable data management platforms capable of high-throughput ingestion and low-latency querying. Recent work has explored the integration of cloud-native data pipelines and edge computing for improving EEWS responsiveness in resource-constrained regions [21]. Additionally, emerging approaches in federated learning and distributed model training further highlight the importance of scalable and decentralized storage infrastructures to support data-intensive seismic applications [22].

Another critical consideration in the design of EEWS storage systems is data heterogeneity. Seismic datasets encompass a wide array of formats—ranging from waveform recordings and metadata to prediction outputs from machine learning models—which complicates integration and retrieval processes. NoSQL and time-series databases have been increasingly proposed as viable alternatives to traditional relational databases due to their ability to accommodate semi-structured or irregular data schemas [23]. Studies have shown that systems optimized for temporal indexing can significantly reduce retrieval latency, which is vital for alert generation and early-stage situational assessment [24].

Furthermore, performance benchmarking under varying operational conditions is essential for determining the suitability of database systems in seismic applications. Factors such as IOPS, throughput, fault tolerance, and horizontal scalability directly affect system resilience during high-load periods, such as aftershocks or multiple concurrent seismic events. These performance characteristics must be rigorously evaluated using standardized metrics to inform the deployment of robust earthquake early warning infrastructures [25].

This study is directed toward examining the architecture and deployment of a database framework optimized for the storage of earthquake prediction outputs. To this end, three distinct database configurations are evaluated: a standard implementation of MongoDB, a horizontally scalable version of MongoDB using sharding, and InfluxDB, a time-series database optimized for temporal data. Additionally, the MiniSEED (mseed) format is assessed as a baseline file-based storage solution, with its performance and applicability for raw seismic waveform storage compared against the aforementioned database systems.

2. Background and Related Work

2.1. Seismic Activity in Indonesia

The Indonesian archipelago and its surrounding regions are characterized by intense seismic activity, offering valuable insights into the tectonic behavior of this geologically dynamic zone. Positioned at the junction of the Indo-Australian, Philippine Sea, Caroline, and Sunda tectonic plates (Figure 1), the region is subject to complex plate interactions. In its western segment, subduction occurs as the Indo-Australian plate descends northward beneath the Sunda plate along the Sunda-Java trench [7]. Meanwhile, in the eastern part of the country, tectonic convergence involves numerous microplates, forming a deformation belt that encompasses processes such as arc-continent collisions, subduction, strike-slip, thrust, and extensional faulting.

Historical data have indicated that, on average, approximately 18 significant seismic events (with magnitudes ≥ 7.0) take place in this region every ten years, with at least 15 earthquakes of magnitude ≥ 8.0 recorded since the year 1900 [8]. Notably, within the past two decades, five major seismic incidents have been documented, including the catastrophic Great Sumatra-Andaman earthquake and ensuing tsunami on December 26, 2004 [9].

The occurrence and causation of earthquakes at varying depths have been observed to differ according to geographical context. Typically, the frequency of seismic events tends to decline with increasing depth; however, a resurgence in activity is often identified within the mantle transition zone (MTZ). This phenomenon is generally attributed to elevated pressure and temperature conditions at those depths [10], aligning with global patterns observed in earthquake-depth distributions. The uptick in seismicity within the MTZ has been hypothesized to stem from resistance associated with phase transitions occurring at approximately 410, 520, and 660 kilometers in depth [10], [11].

Regions such as the Celebes and Banda Sea in eastern Indonesia exhibit the highest levels of seismicity. A westward progression from Java to Sumatra reveals a reduction in the frequency of deep earthquakes, a pattern likely influenced by variables such as altered plate motion vectors, the presence of younger lithospheric material, reduced subduction velocities, and increased mantle temperatures. In contrast, areas including the Manokwari trough and the western New Guinea trench demonstrate weaker seismic zones, which may be indicative of underdeveloped fault structures. The presence of seismic gaps in these regions may relate to slab tearing or other unresolved geodynamic mechanisms. Earthquakes occurring in these zones, often exceeding magnitude 7.0, pose substantial seismic hazards—as exemplified by the 2009 Mw 7.6 Padang earthquake [7].

Indonesia's unique tectonic configuration, shaped by the convergence of both major and micro tectonic plates, renders it one of the most seismically volatile areas on the planet. The consistent occurrence of high-magnitude events, particularly in locales such as the Celebes and Banda Sea, reinforces the critical need for robust seismic monitoring systems and comprehensive risk evaluation strategies. Moreover, the seismic variability across regions and the distinctive dynamics within the mantle transition zone underscore the complexity of tectonic processes in the area. A deep understanding of these geophysical phenomena is essential for enhancing earthquake forecasting capabilities and reducing the impact of large-scale seismic events.

2.2. QuakeFlow System

QuakeFlow represents a state-of-the-art system for seismic monitoring that capitalizes on machine learning techniques and cloud-native infrastructure to improve the identification and classification of seismic events [12]. The workflow encompasses essential procedures such as seismic phase picking and association—tasks vital for discerning earthquake signals and estimating source parameters. Within this framework, sophisticated machine learning models, including PhaseNet [13] and GaMMA [14], are utilized. These models are deployed via Docker containers and orchestrated in a scalable Kubernetes environment.

PhaseNet, based on a convolutional neural network (CNN) architecture, is employed to predict the arrival times of P- and S-waves from seismic recordings with high precision. GaMMA, an unsupervised learning algorithm based on Gaussian Mixture Model Association, probabilistically clusters seismic phase arrivals to estimate earthquake locations and magnitudes. By integrating these models, QuakeFlow has demonstrated superior performance over traditional rule-based algorithms, thereby significantly enhancing both the accuracy and efficiency of seismic event detection. The modular design of the system also permits the seamless integration of additional or updated models, ensuring its adaptability to future technological advancements in the domain of earthquake monitoring.

The QuakeFlow framework supports both batch-oriented and streaming processing modes. Historical seismic datasets are processed concurrently through Kubernetes and Kubeflow pipelines, enabling efficient data exploration and large-scale analytics. For real-time applications, seismic waveform data and inference results are streamed using Apache Kafka. These streams are processed by Spark Streaming, which performs extraction, transformation, and loading (ETL) operations before forwarding the preprocessed data to PhaseNet and GaMMA for analysis. The prediction outputs are then stored in a MongoDB database and disseminated through Kafka to a web-based visualization interface, thereby facilitating near-instantaneous earthquake detection and reporting capabilities [12].

3. Methodology

3.1. Earthquake Data Acquisition

This investigation centers on seismic waveform data that were acquired through the use of the GEOFON (GFZ) seismic client, with a specific emphasis on the Garut earthquake event that transpired in September 2024. The dataset utilized comprises seismic recordings captured on September 18, 2024, within a continuous 24-hour window ranging from 00:00 to 23:59 UTC. The spatial extent of the study focuses on coordinates proximate to Garut, Indonesia (latitude −7.19°, longitude 107.67°), encompassing both latitudinal and longitudinal coverage of 50 degrees, thereby enabling the capture of regional seismicity over a broad geographic range.

The dataset includes waveform data from the Global Seismographic Network (GE), which maintains seismic monitoring capabilities across Indonesia. Seismic stations—such as BBJI, located in Bungbulang, Garut (Java)—played a pivotal role in the collection of this data. Seismograms corresponding to the vertical (Z), north-south (N), and east-west (E) components were retrieved via the BHZ, BHN, and BHE channels, respectively. These

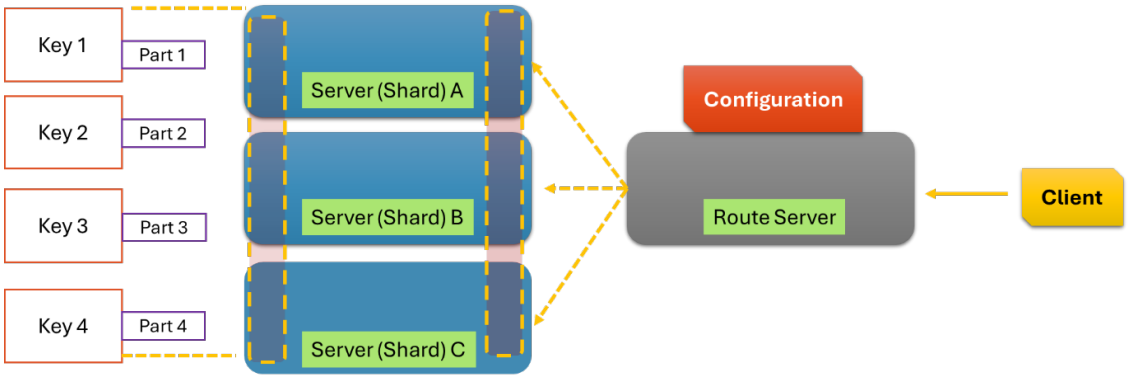

Figure. 1. The architectural design of MongoDB sharding comprises shard units, routing mechanisms, and configuration server components [18].

components collectively yield a comprehensive representation of ground motion, thereby facilitating in-depth analysis of the seismic event.

To ensure uniformity and computational suitability, the waveform data underwent preprocessing procedures. These included the resampling of all seismic traces to a standardized frequency of 100 Hz, thereby maintaining temporal consistency across the dataset. The waveform signals were subsequently transformed into NumPy array format to enhance efficiency in storage and facilitate seamless integration into machine learning workflows and advanced analytical pipelines. This data preparation enables reliable execution of earthquake detection and forecasting tasks.

3.2. Database Models

MongoDB is classified as a NoSQL database and is document-oriented in nature. It utilizes a flexible binary JSON format (BSON) for data representation. Due to its schema-less structure, it allows each document to possess an independent structure, thus accommodating evolving data models and facilitating the storage of unstructured or semi-structured data. This adaptability renders MongoDB particularly effective for applications that necessitate the management of heterogeneous and complex data types [15].

MongoDB's sharding technique is a method implemented for horizontal scaling, wherein datasets are segmented into smaller partitions known as shards, and these are distributed across multiple database instances or clusters [16]. Each shard is responsible for maintaining a subset of the overall data, and collectively, they comprise the entire dataset (Figure 1). Through this partitioning mechanism, MongoDB's capacity for high data throughput and parallel processing is enhanced, making it more suitable for handling extensive datasets across distributed systems [17]. The single-node MongoDB configuration was deployed without replication. This setup aimed to assess baseline performance without the influence of replica set overhead.

InfluxDB is an open-source database designed specifically for time-series data. It is engineered to manage high-ingestion rates and to efficiently query large-scale temporal datasets. Its architecture is particularly suitable for use cases such as monitoring systems, sensor networks, Internet of Things (IoT) applications, and analytic services. Key features of InfluxDB that differentiate it from traditional data management systems include built-in lifecycle management for multiple records, summary statistics processing, and support for wide-range temporal data scans [19].

In the Quakeflow system, we configured MongoDB with the architecture set to "replicaset," which enables native replication across multiple nodes. This setup, without sharding, ensures high availability and data redundancy by maintaining synchronized copies of the database. The replica set configuration serves as the ground truth for the system, providing a reliable and consistent data source that supports fault tolerance and seamless failover in case of node failures.

MiniSEED is a compact binary data format that is predominantly employed in the field of seismology for the storage of continuous waveform recordings. The format is optimized for efficient storage and rapid exchange of seismic data across institutional networks and research entities. Although the data are encoded in binary, they can be decoded and translated into interpretable formats, facilitating the identification of

seismic phases (e.g., P-waves and S-waves). This allows researchers to effectively analyze and interpret seismic activity through data conversion into human-readable formats.

3.3. Evaluation Metrics

The performance of the various database configurations in this study is assessed based on their capability to store and retrieve data efficiently. Two key performance metrics are employed; Input/Output Operations Per Second (IOPS) serves as a metric to evaluate the responsiveness and processing efficiency of a database system under conditions of intensive read and write operations. This is particularly relevant in the context of seismic monitoring, where high-volume, real-time data interactions are critical. Comparative analysis of IOPS across the selected database models provides insights into their operational viability for time-sensitive applications in earthquake forecasting and emergency management. The metric is defined mathematically as follows:

$$IOPS = \frac{\text{Total Number of I/O Operations}}{\text{Time Taken for Operations (in seconds)}} \quad (1)$$

Here, the total number of I/O operations encompasses all read and write activities executed by the database during the monitoring period.

Throughput (Thr) quantifies the rate at which data are transmitted or processed over a given time interval. It reflects the database's efficiency in managing large volumes of data and is particularly important for high-throughput environments such as seismic data ingestion and processing. It is expressed using the following formula:

$$Thr = \frac{\text{Total Data Transferred (in kilobytes)}}{\text{Time Taken (in seconds)}} \quad (2)$$

This metric provides an additional perspective on database suitability by measuring sustained data processing capacity under varying load conditions.

IOPS and throughput were measured using Python scripts instrumented with the time and psutil libraries. I/O operations were tracked at the file system level, and throughput was calculated using wall-clock time and file sizes. The scripts used are available at:

Table 1. provides an overview of the experimental infrastructure, detailing the computational resources allocated for each database configuration, including the number of nodes, CPU cores, memory per node, storage type, and regional deployment within the Google Kubernetes Engine (GKE).

Table 1. Infrastructure specifications (per node).

Specification	Descriptions
Nodes	1 for single-node DB, 3 for MongoDB Sharding
vCPU	2 (N1-standard-2)
RAM	7.5 GB per node
Disk	SSD (Persistent Disk, 100 GB)
Region	GKE Region: us-central1-a

4. Experiments and Results

All experimental procedures were executed within a cloud-native environment utilizing a Google Kubernetes Engine (GKE) cluster configured with N1-Standard-2 virtual machines and SSD storage, selected to maximize computational performance. With the exception of MiniSEED (mseed), each database model was integrated with the QuakeFlow machine learning platform, which employs PhaseNet [13] and GaMMA [14] models to produce earthquake prediction outputs. These outputs were subsequently used to evaluate each database's capability to store and retrieve prediction results, particularly under conditions involving substantial seismic data volumes produced by the GaMMA model.

Performance evaluation was conducted using two metrics: Input/Output Operations Per Second (IOPS) and Throughput. These metrics were recorded using two distinct data subsets—242 rows, representing approximately one hour of prediction output, and 845 rows, corresponding to six hours of model inference.

The two subsets represent 1-hour and 6-hour prediction outputs respectively, to simulate different processing intervals in near real-time systems. Both are temporally consecutive and differ in volume, not in schema. Unlike the database models, the mseed format was not used for predictive analysis; instead, it was tested based solely on reading, writing, and downloading waveform data through the ObsPy library [20].

In order to establish a baseline, each storage model was tested without the inclusion of performance optimizations or system tuning. The intent of this approach was to assess the inherent or "raw" capabilities of each database system with respect to data insertion, retrieval, and scalability when processing large-scale seismic datasets.

4.1. IOPS Results

The first metric considered in this evaluation was Input/Output Operations Per Second (IOPS), which quantifies the responsiveness and capacity of a database to manage concurrent read and write operations.

MongoDB demonstrated stable and scalable performance across both datasets. IOPS values increased from 1763.0 for 242 rows to 1826.9 for 845 rows, suggesting that the system maintained its efficiency as the data volume increased. This indicates MongoDB's potential for consistent performance even under extended workloads.

In contrast, MongoDB with sharding showed reduced operational efficiency. Although this configuration was intended to enhance scalability, the measured IOPS values dropped slightly, from 1066.7 (242 rows) to 1055.3 (845 rows). These results imply that the additional overhead introduced by sharding mechanisms may adversely impact performance in environments with moderate data volumes.

InfluxDB, a time-series optimized database, recorded superior IOPS performance compared to both MongoDB configurations. It achieved 1247.7 and 1301.4 IOPS for the 242-row and 845-row datasets, respectively. The slight improvement in IOPS with increased data size indicates that InfluxDB scales efficiently and is highly suitable for applications involving continuous data ingestion.

On the other hand, the mseed format exhibited the poorest performance in terms of IOPS. A dramatic decline was observed—from 170.3 IOPS with the smaller dataset to just 17.7 with the larger one. These results reflect significant limitations in the format's ability to handle larger datasets, particularly when rapid access to waveform segments is required.

4.2. Throughput Results

The second performance metric examined in this study was Throughput, which measures the volume of data a system can process over a given time interval, expressed in kilobytes per second (KB/s). This metric offers insights into a system's capacity for high-volume data transfer.

MongoDB demonstrated moderate throughput that increased proportionally with data volume. It processed 72.52 KB/s for 242 rows and improved to 98.85 KB/s with 845 rows. These findings confirm MongoDB's ability to handle progressively larger datasets while maintaining acceptable processing speeds.

Meanwhile, MongoDB Sharding displayed a relatively stable throughput performance. For the smaller dataset, throughput was measured at 77.23 KB/s, slightly decreasing to 71.41 KB/s for the larger dataset. Although performance was comparable to the standard MongoDB setup, the marginal decline suggests a possible trade-off between horizontal scalability and throughput consistency.

Table 2. Input/Output Operations Per Second (IOPS) and Throughput Performance (KB/s) across storage models.

Database Model	Data Quantity (Rows)	IOPS	Throughput (KB/s)
MongoDB	242	1763.0	72.52
	845	1826.9	98.85
MongoDB Sharding	242	1066.7	77.23
	845	1055.3	71.41
InfluxDB	242	1247.7	71.98
	845	1301.4	75.18
MiniSEED	242	170.3	7,409,620.16
	845	17.7	4,725,019.19

The variation in MongoDB performance is attributed to internal caching and index optimization that disproportionately benefits smaller data sizes. In contrast, MongoDB Sharding introduces coordination overhead that stabilizes performance but adds latency.

As describe in Table 2, InfluxDB sustained a throughput of 71.98 KB/s for 242 rows and slightly improved to 75.18 KB/s with 845 rows. The consistency in its performance reinforces its suitability for time-series applications where stable data flow and ingestion rates are essential.

In contrast, the mseed format achieved extraordinarily high throughput values—7,409,620.16 KB/s and 4,725,019.19 KB/s for the smaller and larger datasets, respectively. These exceptional figures are largely attributed to the inherently large file sizes of mseed format, which enables bulk data to be transferred rapidly. However, this high throughput does not imply suitability for real-time analysis or querying, as mseed lacks the flexibility and indexing features of database systems.

Although MiniSEED is structurally distinct from modern distributed databases, its inclusion in this study is both deliberate and essential. As the de facto standard format in geophysics for storing waveform data, including from global and regional networks such as GEOFON, MiniSEED represents the raw data structure most commonly encountered in seismic applications. By comparing NoSQL databases like MongoDB and InfluxDB against MiniSEED, this study evaluates how modern storage technologies can complement or potentially substitute traditional flat-file formats within real-time EEWS

pipelines. This comparison serves to bridge the gap between legacy geophysical data standards and contemporary database architectures, highlighting not only their performance differences but also their integration potential in operational seismic monitoring systems.

4.3. Implications for EEWS

MiniSEED offers high throughput due to its compact and straightforward format, which eliminates the need for complex schemas. This design allows for rapid transmission of large volumes of data, prioritizing data transfer efficiency over detailed, operation-heavy processing. However, while MiniSEED is highly effective for transporting seismic data, it requires additional processing to decode and convert the raw binary format into a human-readable and informative representation suitable for analysis.

The experimental results highlight the comparative strengths and weaknesses of each storage solution with respect to IOPS and throughput performance. Both MongoDB and InfluxDB demonstrated promising capabilities for real-time data processing, suggesting their appropriateness for deployment within Earthquake Early Warning Systems (EEWS).

Despite the encouraging performance metrics, it should be noted that the current evaluation was confined to a specific hardware environment—namely N1 Standard-2 instances on Google Kubernetes Engine. Further research is warranted to investigate these systems under varied hardware configurations and stress conditions in order to assess their maximum operational capacities.

In the context of real-time applications such as Earthquake Early Warning Systems (EEWS), selecting the appropriate performance metric depends on the characteristics of the data workload. For systems that require rapid ingestion and immediate response to numerous small, time-sensitive updates such as seismic event detection and alert triggering IOPS is the primary concern, as it directly affects latency and the ability to handle high-frequency, concurrent operations.

Conversely, in scenarios involving continuous ingestion of large seismic waveform streams or AI-driven inference results, sustained throughput becomes a more relevant metric to ensure the system can accommodate high data volumes without bottlenecks. While a moderate correlation between IOPS and throughput was observed in this study, storage formats like MiniSEED highlight that high throughput alone does not guarantee responsiveness due to their limited query capabilities. Therefore, EEWS architectures should prioritize low-latency designs with sufficient IOPS to support real-time alerting, and then scale throughput accordingly to handle growing data volumes without introducing buffering delays or processing lag.High IOPS and throughput are essential for ensuring system scalability, availability, and responsiveness, all of which are fundamental attributes for EEWS platforms operating in seismically active regions. As such, these performance indicators are directly relevant to maintaining real-time alerting mechanisms and facilitating prompt decision-making during seismic events.

Based on the experimental results and the operational demands of Earthquake Early Warning Systems (EEWS), MongoDB and InfluxDB emerge as the most suitable technologies for real-time implementation. MongoDB demonstrated the highest Input/Output Operations Per Second (IOPS) across both small and moderate data volumes, making it well-suited for workloads that involve frequent, time-sensitive transactions such as alert generation and metadata updates.

InfluxDB, on the other hand, exhibited more consistent scalability and efficient throughput handling, which is advantageous for continuous seismic data streams and time-series queries common in EEWS pipelines. While MongoDB with sharding provides a theoretical path for horizontal scaling, our results show that the overhead introduced significantly reduced IOPS performance in moderate-scale deployments.

Additionally, although the MiniSEED format achieved exceptionally high throughput, its limited query capabilities and poor responsiveness under larger data sizes render it unsuitable for real-time analytics. Therefore, considering the balance between low-latency access, scalability, and throughput performance, MongoDB is preferable for metadata-centric, high-frequency operations, while InfluxDB is more appropriate for managing large-scale, temporally indexed waveform data within an integrated real-time EEWS architecture.

Additional long-term evaluations should be conducted to determine the sustained reliability, cost-effectiveness, and adaptability of these storage architectures under fluctuating network conditions and expanding data volumes. The insights gained from such investigations could inform the future design and optimization of robust, high-performance infrastructures for earthquake monitoring and disaster response.

5. Conclusion

This study aimed to evaluate the performance of different data storage systems—MongoDB, MongoDB with sharding, InfluxDB, and MiniSEED for managing real-time seismic prediction data in Earthquake Early Warning System (EEWS) applications. The evaluation was conducted using benchmark metrics focused on Input/Output Operations Per Second (IOPS) and data throughput, under a controlled cloud-native environment. The results demonstrated that MongoDB and InfluxDB offer strong performance for real-time data workloads, with MongoDB achieving higher IOPS and InfluxDB exhibiting better scalability with increasing data volume. In contrast, MongoDB with sharding introduced system overhead that reduced its efficiency, and MiniSEED, although showing high throughput due to its flat-file nature, lacked responsiveness and was unsuitable for real-time analytics. Future research should explore these systems using high-volume datasets, real-world EEWS deployment conditions, and include comparisons with modern data lake architectures to better assess their operational robustness in seismic risk mitigation efforts.

References

[1] R. M. Allen, ‘‘Rapid magnitude determination for earthquake earlywarning,’’ in The Many Facets Seismic Risk, vol. 4. Naples, Italy:Università degli Studi di Napoli ‘Federico II’, 2004, pp. 15–24.

[2] R. M. Allen, ‘‘Probabilistic warning times for earthquake ground shakingin the San Francisco bay area,’’ Seismol. Res. Lett., vol. 77, no. 3,pp. 371–376, May 2006, doi: 10.1785/gssrl.77.3.371.

[3] R. M. Allen, ‘‘The ElarmS earthquake early warning methodology andapplication across California,’’ in Earthquake Early Warning Systems.Berlin, Germany: Springer, pp. 21–43, 2007.

[4] S. Tunç, B. Tunç, D. Çaka, Ş. Barış, Dünyada yaygın olarak kullanılan erken uyarı sistemleri, 6th International Conference on Earthquake Engineering and Seismology, Kocaeli, pp. 895–909, 2021.

[5] W. Zhu et al., “QuakeFlow: a scalable machine-learning-based earthquake monitoring workflow with cloud computing,” Geophysical Journal International, vol. 232, no. 1, pp. 684–693, Sep. 2022, doi: 10.1093/gji/ggac355.

[6] N. J. Lindsey and E. R. Martin, “Fiber-Optic seismology,” Annual Review of Earth and Planetary Sciences, vol. 49, no. 1, pp. 309–336, Jan. 2021, doi: 10.1146/annurev-earth-072420-065213.

[7] S. J. Hutchings and W. D. Mooney, “The seismicity of Indonesia and tectonic implications,” Geochemistry Geophysics Geosystems, vol. 22, no. 9, Sep. 2021, doi: 10.1029/2021gc009812.

[8] U.S. Geological Survey, "Advanced National Seismic System (ANSS) comprehensive earthquake catalog (ComCat)," 2020. [Online]. Available: https://earthquake.usgs.gov. [Accessed: Dec. 23, 2024].

[9] T. Lay et al., “The Great Sumatra-Andaman earthquake of 26 December 2004,” Science, vol. 308, no. 5725, pp. 1127–1133, May 2005, doi: 10.1126/science.1112250.

[10] Z. Zhan, “Mechanisms and implications of deep earthquakes,” Annual Review of Earth and Planetary Sciences, vol. 48, no. 1, pp. 147–174, Dec. 2019, doi: 10.1146/annurev-earth-053018-060314.

[11] M. I. Billen, “Deep slab seismicity limited by rate of deformation in the transition zone,” Science Advances, vol. 6, no. 22, May 2020, doi: 10.1126/sciadv.aaz7692.

[12] W. Zhu et al., “QuakeFlow: a scalable machine-learning-based earthquake monitoring workflow with cloud computing,” Geophysical Journal International, vol. 232, no. 1, pp. 684–693, Sep. 2022, doi: 10.1093/gji/ggac355.

[13] W. Zhu and G. C. Beroza, “PhaseNet: a Deep-Neural-Network-Based seismic arrival time picking method,” Geophysical Journal International, Oct. 2018, doi: 10.1093/gji/ggy423.

[14] W. Zhu, I. W. McBrearty, S. M. Mousavi, W. L. Ellsworth, and G. C. Beroza, “Earthquake phase Association using a Bayesian Gaussian mixture model,” Journal of Geophysical Research Solid Earth, vol. 127, no. 5, Mar. 2022, doi: 10.1029/2021jb023249.

[15] D. Chauhan and K. Bansal, "Using the advantages of NoSQL: A case study on MongoDB," Int. J. Recent Innov. Trends Comput. Commun., vol. 5, pp. 90-93, 2017.

[16] Y. Liu, Y. Wang and Y. Jin, "Research on the improvement of MongoDB Auto-Sharding in cloud environment," 2012 7th International Conference on Computer Science & Education (ICCSE), Melbourne, VIC, Australia, pp. 851-854, 2012 doi: 10.1109/ICCSE.2012.6295203.

[17] Sasikala, R. "Speedy Data Analytics through Automatic Balancing of Big Data in MongoDB Sharded Clusters." In Computational Intelligence Applications in Business Intelligence and Big Data Analytics, pp. 179-210, 2017.

[18] H. Shin, K. Lee, and H.-Y. Kwon, "A comparative experimental study of distributed storage engines for big spatial data processing using GeoSpark," The Journal of Supercomputing, vol. 78, pp. 1-24, 2022, doi: 10.1007/s11227-021-03946-7.

[19] M. Nasar and M. A. Kausar, "Suitability of InfluxDB database for IOT applications," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 10, pp. 1850–1857, Aug. 2019, doi: 10.35940/ijitee.j9225.0881019.

[20] M. Beyreuther, R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann, "OBSPY: a Python toolbox for seismology," Seismological Research Letters, vol. 81, no. 3, pp. 530–533, May 2010, doi: 10.1785/gssrl.81.3.530.

[21] B. Ghosh, P. Kumar, and N. Patra, "Cloud-based architecture for scalable earthquake early warning systems," IEEE Access, vol. 10, pp. 75811–75822, 2022, doi: 10.1109/ACCESS.2022.3189110.

[22] J. Liu, H. Lin, and S. Zhang, "Federated learning for distributed seismic signal analysis," Seismological Research Letters, vol. 93, no. 4, pp. 2171–2180, 2022, doi: 10.1785/0220210418.

[23] K. M. Lee and A. Gupta, "Time-series database selection for real-time geophysical data analysis," Geoscience Data Journal, vol. 9, no. 3, pp. 435–449, 2022, doi: 10.1002/gdj3.164.

[24] T. R. Henderson, R. M. Allen, and M. Kohler, "Low-latency seismic data retrieval using time-indexed NoSQL databases," Bulletin of the Seismological Society of America, vol. 111, no. 5, pp. 2315–2326, 2021, doi: 10.1785/0120200379.

[25] F. Montazeri, S. Sadighian, and M. Akbari, "Benchmarking NoSQL databases for geophysical data applications," Journal of Applied Computing and Geophysics, vol. 4, no. 1, pp. 45–57, 2023, doi: 10.1016/j.jacg.2023.01.005.