Using web enables model for real-time ecological data warehouse

Klaokanlaya Silachan[1*] and *Ouychai* Intrarasombat[1]

[1]Computer technology Program, Faculty of Science and Technology, Nakhon Pathom Rajabhat University, 85 Malaiman Road, Muang, Nakhon Pathom, 73000, Thailand

Abstract: This research involves the development of a web model to transform data from an ecology dataset, specifically climate data, into a non-relational database format (NoSQL) in the form of a real-time document-based database. It consists of MongoDB nodes and CouchDB, comparing the speed of reading data from the database to select the best-performing non-relational database for managing data values in the database. It is in a document-based format. It tests by loading the first 100 data sets, followed by 200, 300, 400, 500, 600, 800, and 1000 data sets. The experimental results show that the MongoDB server model has the best performance with an average value of 2,190,691.

1 Introduction

The database is a type of data management system that is large in size, capable of collecting data from various sources over multiple time periods, and accumulating large amounts of historical data to support efficient data retrieval by users according to the database structure, which differs from general databases. For example, storing data for use in an organization's data warehouse software. Generally, long-term data storage is conducted for at least 5-10 years for analysis [1-12].

Currently, the development of databases has evolved into NoSQL structured formats, specifically semi-structured data or document-based databases, which have become of interest for development into non-relational database formats. This allows databases to be managed by document-oriented NoSQL systems. Software for managing non-structural data in a document-based format includes several options such as MongoDB, CouchDB, and others (Jarosław, 2017).

If the real-time processed database can accept data from OLTP systems immediately, the data will be sent from the main database to the data warehouse, supporting efficient processing for simultaneous data retrieval and analysis. It is necessary to develop a system to process data in real-time by coding into the data software instead of batch or offline data conversion methods. This helps reduce the time required for loading data during the data transformation process, enabling immediate data analysis and selection of database formats suitable for efficient data loading and processing to generate reports for further analysis. Generally, structured data is stored in databases with rigorous data frameworks. If changes

* Corresponding author: klaokanlaya@gmail.com

are necessary, all structured data must be updated, which consumes a significant amount of time and resources [5].

For systems designed to manage data in a database format, one interesting aspect is the development of systems for ecological systems, which have been developed into database formats for the benefit of future data analysis. One such aspect is climate data, which includes data on climate change, such as changes in average weather conditions and any related changes, which results in increasing data volumes daily. Organizations or datasets that have collected data in this database format are diverse [2].

Designing or modifying the structure and climate data to be stored in a non-relational database (NoSQL) will significantly increase data management, access, and reading speeds, as well as predicting future data trends, which will also increase big data.

This research proposes developing system model in web-enabled format to convert data into a non-relational database structure in real-time. It manages data values in the database, comprising MongoDB nodes and CouchDB, to compare the efficiency and speed of reading and fetching data from the database, to select the most efficient database for developing web-enabled database management systems for real-time non-relational databases.

2 Literature Reviews

The Enterprise Data Warehouse (EDW) implementation is most effective when employing up-to-date data modeling techniques and best practices that have been refined over the years, serving as the foundation for data warehousing globally. According to Leonard (2011), when comparing query results, it becomes evident that data retrieval is significantly faster from the organized star schema in the data warehouse compared to the transactional database.

The implementation of data warehousing using Capture, Transform, and Flow (CTF) technology for real-time replenishment is crucial to support heterogeneous environments. Due to technological advancements, corporate mergers, and acquisitions, most organizations now operate multiple computing platforms and databases, each storing separate sets of information that may be incompatible with each other (Vandermay, 2001). Integration teams require real-time data integration with minimal or no data latency for various use cases.

While this whitepaper primarily focuses on data warehousing, it's essential to distinguish between different areas. Various architectures have been utilized to collect transactional data from operational sources to populate data warehouses. These techniques differ mainly in the latency of data integration, ranging from daily batches to continuous real-time integration (Kotopoulis, 2012).

According to Bouaziz et al. (2017), a Real-Time Data Warehouse enables the storage of data at the time of production, immediately capturing, cleaning, and storing it within the data warehouse's structure. The operations occurring backstage of the data warehouse architecture are generally referred to as Extraction, Transformation, and Loading (ETL) processes (Vassiliadis & Simitsis, 2014).

A more practical approach involves a semi-automated environment, where user requests for data freshness and completeness are balanced against the workload of all involved subsystems of the warehouse (sources, data staging area, warehouse, data marts). This approach enables a tunable, regulated flow of data to meet resource and workload thresholds set by the administrators of the involved systems. The significance, complexity, and criticality of such an environment make near real-time warehousing a significant topic of research and practice (Vassiliadis & Simitsis, 2014).

3 Research Methods

3.1 Data Exploration

By using the dataset for research experimentation from climate data Kaggle (https://www.kaggle.com/datasetsgoyaladiclimate-insights-dataset), which provides valuable in-depth information regarding ongoing climate change. This dataset covers the collection of temperature records, CO2 emission data, and comprehensive measurements of rising sea levels, focusing on global level trends. CSV format datasets are often considered semi-structured data, although they may have well-structured layers. Generally, CSV data formats are used to load data into databases, which are non-relational. See Figure 1 for the structure".

Date	Location	Country	Temperati	CO2 Emiss	Sea Lev	Precipitati	Humidity	Wind Spe
1/01/2543 12:00 AM	New Williamtowr	Latvia	10.68899	403.1189	0.7	13.83524	23.631	18.49
1/01/2543 8:09 PM	North Rachel	South Africa	13.81443	396.6635	1.	40.97408	43.9829	34.2493
2/01/2543 4:19 PM	West Williamlanc	French Guiana	27.32372	451.5532	-0.16078	47.89793	96.6526	34.12426
3/01/2543 12:29 PM	South David	Vietnam	12.30958	422.405	-0.47593	5.1	47.46794	8.554563

Fig. 1: Data structure

3.2 System Analysis

This step involves analyzing the scope of the system based on the studied system, with a structure and guidelines for system development and testing.

One of the most convenient options for CSV loading to a destination data warehousing system is to develop a web-based process to transform data into a non-relational database format. There are two options: MongoDB and CouchDB, as shown in Figure 2, to assess the efficiency of response time for data conversion into an efficient database.

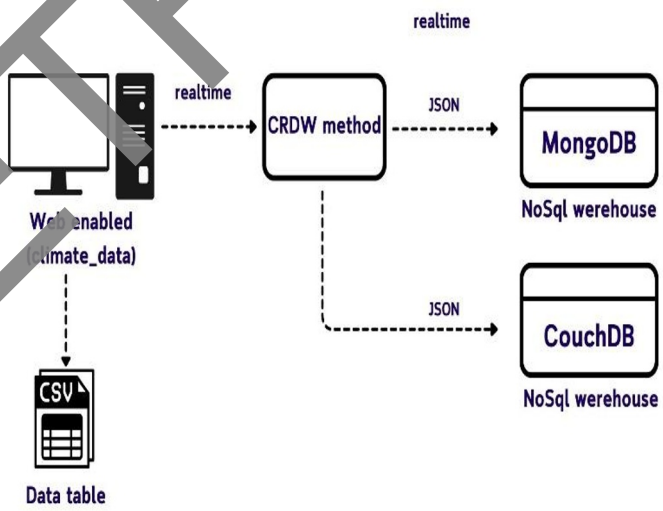

Fig. 2: Structural diagram of the development of a data transformation system into a non-relational data warehouse

3.3 System Design

In system design, a model for the overall system has been developed. In this section, the steps of the data transformation model in real-time non-relational data warehouse can be demonstrated.

The steps for converting data in Climate real-time warehouse (CRDW method) consist of 4 steps as follows:

Step 1: Web-Enable for Data Input - Step 1.1: Enter data through web forms or import datasets. - Step 1.2: Save data into a datasheet in CSV file format. - Step 1.3: Go to Step 2.

Step 2: Convert and Transform Data - Step 2.1: Read and convert data from JSON file. - 2.1.1: Read the file Climate_change_data.csv. - 2.1.2: Check the file type CSV or table. - 2.1.3: Read all data into an array. - 2.1.4: Encode data into JSON format and set data values according to the Time Stamp. - 2.1.5: Display results and save in JSON format.

Step 3: Load Data to MongoDB, Cassandra using Node.js - Step 3.1: Establish connection using node server.js. - Step 3.2: Connect to MongoDB. - Step 3.3: Save data into MongoDB. - Step 3.4: Establish connection and load data using cbdocloader and streamloadjson for Clouch DB. - Step 3.5: Save JSON data into ClouchDB.

Step 4: Load Test for Performance Comparison.

3.4 System Development

The system development for data transformation and management into a data warehouse is carried out using PHP and Node.js languages for converting data into JSON document base format for non-relational databases, including MongoDB and CouchDB.

3.5 System Testing and Evaluation

This system is capable of processing, transforming, and retrieving data from the data warehouse to be presented accurately and in a timely manner later on.

A comparison of the speed of data conversion between non-relational database formats, namely MongoDB and CouchDB, has been conducted.

The test results provide the best time values, which will be used as the basis for developing a non-relational database system for real-time temperature dataset management.

4 Results

4.1 System Development Results

Experimental Results of Data Transformation

From the experimental results of transforming the dataset into the data warehouse in JSON format, the details are as follows in the JSON document structure. This structure is designed to store data in a non-relational database with document features similar to recording transaction timestamps, as shown in Figure 3

Fig. 3: Example of transforming data into a JSON document structure for storing data in a non-relational database format.

4.2 Performance Evaluation Results

For this research, the performance evaluation results are presented based on the average processing time (Load test Time) between NoSQL Models developed in Document-Oriented format and utilizing MongoDB, compared to CouchDB.

Regarding the software and hardware used in the performance evaluation, the database management software includes MongoDB version 3.2 and MySQL 9.5. The hardware specifications used for experimentation are an Intel Core i7 8700 (6 cores / 12 threads), 16GB of RAM, Western Digital Black M.2 Solid State Drive, 10/100/1000 Gbps Ethernet, and Microsoft Windows 11.

The programming languages used for development are PHP and Node.js. The experiments were conducted using climate data datasets for weather conditions.

Table 1: Data Load Test Time (Seconds)

Number of Record Loaded	JSON (MongoDB)	JSON (CouchDB)
200	2.168611	3.391685
400	2.190497	3.544282
600	2.194740	3.675656
800	2.198454	3.820781
1000	2.201153	3.901095
Average	2.190691	3.6667

From the experimental results, it is evident that the development of a web-enabled database in non-relational (NoSQL) format, focusing on document-based storage, has been successful. A comparison of response times within the MongoDB system shows higher efficiency, with an average response time of 2.190691 compared to the creation and storage in CouchDB with an average response time of 3.6667. This is because MongoDB reads and loads data faster, resulting in faster system performance compared to other methods.

System Development Results for Real-Time Web-Enabled Database

In developing a web-enabled system for Climate data, to implement the CRDW method for data transformation into the MongoDB and CouchDB databases, data management through form submission, either individually or as datasets, has been facilitated. The data is stored in a datasheet, divided into two parts: Climate Data section and data transformation into a non-relational database (MongoDB NoSQL warehouse).

5 Discussion and Conclusion

This research involved the design and development of a web-enabled platform and methodology for transforming data from climate data sets into a real-time non-relational (NoSQL) database focusing on document-based storage.

In the study, tests were conducted to compare the processing speed of data retrieval from both MongoDB and CouchDB databases. It was found that MongoDB outperformed CouchDB in terms of efficiency. The tests involved processing the first 200 datasets, followed by 400, 600, 800, and finally 1000 datasets. MongoDB, in conjunction with the designed methodology and storage structure, demonstrated faster data retrieval times compared to CouchDB. On average, MongoDB took less time to process data compared to CouchDB.

Therefore, MongoDB, as a non-relational database, proved to be capable of processing data swiftly. The results indicated that MongoDB is the most efficient database system for this research project due to its faster data retrieval speed and overall performance, making it suitable for use as a data warehouse.

References

1. Abdelhedi, F., Jemmali, R., Zurfluh, G. (2022). Relational Databases Ingestion into a NoSQL Data Warehouse.
2. Agapito, G., Zucco, C., Cannataro, M. (2022). *COVID-warehouse: a data warehouse of Italian COVID-19, Pollution, and Climate Data.* International Journal of environmental research and public health.

3. Bouaziz, S., Nabli, A., Gargouri, F. (2017). From Traditional Data Warehouse To Real Time Data Warehouse: Advances in Intelligent Systems and Computing.
4. Harvy, I., Matitaputty, G., Girsang, A., Michael, S., Isa, S. (2019). *The use of book store GIS data warehouse in implementing the analysis of most books selling.* Proceedings of the 7th International Conference on Cyber and IT Service Management (CITSM), 1-5.
5. Hassan, C.A., Hammad, M., Uddin, M., Iqbal, J., Sahi, J., Hussian, S., Ullah, S. (2022). IEEE Access, **10,** 13472-13480.
6. Kotopoulis, A. (2012). Best Practices for Real-time Data Warehousing. Oracle Corporation World Headquarters.
7. Kurpanik, J. (2017). Journal of Science of the Military Academy of Land Forces, **49**, 3(185).
8. Leonard, E. (2011). Design and Implementation of an Enterprise Data Warehouse. Master's Theses.
9. Lukić, J.J..(2014). Approach to Multidimensional Data Modeling in BI Technology. ICIST 2014 (2).
10. Songsiriand, K., Tamee, K. (2022). Journal of Applied Informatics and Technology, **4(2):** 99-113.
11. Vanderma, J. (2001). Considerations for Building a Real-time Data Warehouse. Data Mirror Corporation White Paper.
12. Vassiliadis, P., Simitsis, A. (2023). Near Real Time ETL. Available online: https://www.researchgate.net/publication/226219087.