

Original Article

# Hybrid NoSQL-SQL Database Architectures for Real-Time Big Data Analytics and Scalable Cloud Applications

\*Venkata Kishore Chilakapati<sup>1</sup>, Srikanth Reddy Keshireddy<sup>2</sup>, Venkata Teja Nagumotu<sup>3</sup>, Harsha Vardhan Reddy Kavuluri<sup>4</sup>, Akhil Kumar Pathani<sup>5</sup>, Ajay Dasari<sup>6</sup>

<sup>1</sup>Support Escalation Engineer, Microsoft.

<sup>2</sup>Senior Software Engineer, Keen Info Tek Inc.

<sup>3</sup>Sr Network Engineer, Techno-bytes Inc.

<sup>4</sup>Lead database administrator, Wissen infotech Inc.

<sup>5</sup>Network Engineer, Ebay.

<sup>6</sup>Senior Support Engineer, Microsoft.

## Abstract:

The increasing data volume, velocity and variety has been a serious challenge to the conventional data management systems. To overcome scalability, heterogeneity and low-latency processing issues, this paper discusses a hybrid SQL-NoSQL architecture, which combines in-memory databases (MMDB), relational databases (RDB) and distributed file systems (DFS) to manage spatio-temporal big data efficiently. The framework that is proposed allows real-time ingestion of data, on-the-fly processing of analytics, and scalable long-term data storage using a single integrated hybrid framework. The paper also analyzes data interoperability, ACID- BASIS consistency trade-offs, streaming systems/frameworks, e.g., Spark, Flink and Storm and event-driven high-velocity analytics architectures. Also, the use of AI/ML methods in predictive scaling and performance optimization of cloud-based systems is pointed out. Literature review is conducted to establish major gaps in the areas of schema transformation, distributed query optimization, and hybrid consistency management. The results reveal that hybrid SQL-NoSQL systems provide a robust platform of real-time analysis of large amounts of data and scalability cloud applications as well as modern decision-making systems. On the overall, the study presents a thorough architectural framework to the creation of scalable, real-time, big data analytics systems.

## Keywords:

Hybrid Database, Nosql, SQL, Real-Time Data, Scalable Systems, Stream Processing Frameworks, Data Management. Cloud Computing.

## Article History:

**Received: 18.11.2023**

**Revised: 20.12.2023**

**Accepted: 03.01.2024**

**Published: 14.01.2024**

## 1. Introduction

As the technology environment rapidly changes and addresses novel data centric challenges, companies have been forced to consider advanced methods of data management that can account to the needs of dynamic business, massive applications and real time processing needs. Over three decades, the RDBMS has been the standard of data storage in companies, and Structured Query Language (SQL) offers universal data manipulations, a high level of transactional integrity using ACID principles, and a structured schema-based

design [1]. But the rapid increase in the amount of data, its speed and diversity has revealed that there are constraints in the conventional relational structures. SQL-based systems are not very efficient to scalability to high horizontal services and to handle highly heterogeneous and unstructured data and to support distributed cloud-based applications necessary to the modern Big Data analytics [2]. Even though cluster-based computing and other extended SQL functions like A JSON and XML storage options have been added, they are not adequate to satisfy real time and high throughput analytical workloads.

NoSQL (Not Only SQL) data stores came up and managed some of these challenges by providing a flexible schema, horizontal scalability, high availability, and performance optimization under distributed architecture [3]. NoSQL databases, such as document stores (e.g., MongoDB), key-value stores (e.g., Redis), wide-column databases (e.g., Cassandra), and graph databases (e.g., Neo4j), have specific qualities that allow the processing of heterogeneous and unstructured data. Nevertheless, in spite of these benefits, NoSQL systems tend to be weak in regard to strong consistency and complex transactional processing capabilities that SQL databases possess [4].

As a result, hybrid SQL-NoSQL database models are becoming more and more popular in the modern cloud ecosystem environment as well as in real-time applications of Big Data, being the reliable and ACID-compliant relational systems with the scalability and performance attributes of non-relational solutions [5]. The hybrid model would help organizations to combine both structured and unstructured data, optimization, and scalability, fault tolerance and analytics-directed designs that are compatible with next-generation cloud models.

### 1.1. Organization of the Paper

The structure of the paper is as follows The hybrid database architecture is covered in Section II. In section III, the data management and interoperability are discussed. Section IV explains the role of NoSQL systems in Big Data analytics. Section V describes hybrid architectures for real-time analytics, including streaming frameworks and AI-driven scaling. Section VI provides a detailed literature review, and Section VII concludes the study with future research directions.

## 2. Nosql-Sql Dbms Hybrid Storage Architecture

The hybrid database architecture models combine the benefits of NoSQL and SQL systems to provide an efficient system that can manage heterogeneous, large-scale, real-time data. The suggested method's main innovation is to regulate the dataflow of spatiotemporal change information and connect the three databases using the change element. Figure 1 displays the three NoSQL-SQL DBMS types. The following three sub-structures carry out distinct functions:

- The MMDB functions as an access mechanism. It synchronises previous data with the DFS in batches after sanitisation and enables for real-time writing and querying of the current input data.
- The RDB operates as a secondary element. It maintains structured on-the-fly results for analysing and changing information gathered from input data, uses a trigger mechanism to detect abnormal events in real-time, and immediately delivers events for related geographical objects and modules using the "subscribe/publish" message mechanism for dynamic geo-processing.
- The most significant factor is the DFS. It maintains incremental, large-volume historical geographic data, with even distribution and a scalable storage environment.

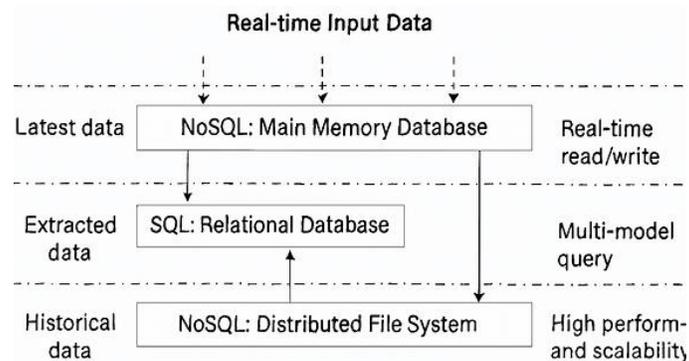


Figure 1. Framework of Hybrid Storage Architecture [6]

### 2.1. MMDB for Real-Time Access

Access efficiency for geoprocessing deteriorates when enormous volumes of real-time geospatial data accumulate because of I/O constraints and the growing requirement for temporal index maintenance. Pre-processing, which includes information extraction, data cleaning, and duplication removal, is also necessary since the original spatiotemporal data contains an uneven distribution of data values and a significant number of incorrect or duplicate data. In other words, only true real-time spatiotemporal data that is not repeated deserves to be serialised [6].

### 2.2. DFS for Incremental Historical Data

The vast volumes of "outdated" spatiotemporal data must be saved on disk in order to support future data mining. With the buildup of incremental data, standard centralized RDB storage solutions suffer from I/O bottlenecks and lack of scalability, making them unsuitable for large-scale spatiotemporal data storage. Commonly used DFS, a prominent type of NoSQL database, feature an extensible design made up of many storage servers that may balance the requirement to schedule and store large amounts of spatiotemporal data.

### 2.3. RDB for On-The-Fly Extracted Data

It is critical to extract useful information ahead of time before serialising geographical data. The interest amount, anomalous change value, and digest value are among the specifics. These three lightweight value types decrease the amount of storage in a record, improve query performance, and extract the value of multi-source heterogeneous input data.

## 3. Data Management and Interoperability

The data volume has grown so fast that it has increased the challenge of handling the diversity and achieving cross-heterogeneous interoperability. Making the transformation and integration of various data structures fully is not feasible. In its place, the new methodologies are based on querying various types and forms of data through the same interface. Different query languages have been developed: relational, object-oriented, hierarchical and many different NoSQL languages, all based on a variety of storage paradigms [7]. Polyglot Persistence and Data Lakes demonstrate the high occurrence of heterogeneous data storage, such that the historical methods of unifying data, such as warehouses, are not applicable given the scale, speed and diversity of Big Data. Interoperability is not accomplished by imposing physical transformation of data but by query translation methods since none of the existing storage or query paradigms is better than the rest. These techniques translate a universal query language into a variety of model-specific languages. They provide for standardised ad hoc access to a wide range of data sources. This does not come at the cost of scalability or expressivity as integration, analytics, and cross-system usage of data are made easy.

### 3.1. Consistency and ACID-BASE Trade-Offs

Consistency is not normally assured to a distributed NoSQL system as it is in a normal relational system. Data Consistency Simulation Tool of NoSQL Database System is a model that characterises the behaviour of the data consistency in the network partitions and replications scenarios. Most NoSQL databases use a BASE (Basically Available, Soft State, Eventual Consistency) policy, which is meant to provide better availability and scale, rather than the same policy as relational databases. The above trade-off implies that increasing the availability and the performance (through BASE) may result in the reduction of the consistency and the adoption of the high consistency would increase the availability or reduce the latency.

### 3.2. Real-Time Data Processing and Low-Latency Access

The most recent techniques for achieving low-latency access and real-time data processing are SOA and CDC [8]. SOA isolates the data consumers and controllers, which permits accessing remote data in real-time and independently. Web services in the form of WSDL or REST are frequently implemented to allow communication between distributed components. Also, streaming real-time applications have been offered under distributed messaging servers such as Apache Kafka which offer low-latency streaming. Kafka features four key APIs: the Producer API for posting data streams, the Consumer API for subscribing to topics, the Streams API for processing and manipulating streams, and the Connector API for connecting to other systems.

### 3.3. Query Federation and Data Virtualization

Query federation and data virtualization allows access to heterogeneous data sources in a single way without the need to physically move and consolidate data. Virtual views are built in memory using distributed query execution and result federation, rather than of the incoming data to a centralized warehouse. The method offers agility to data integration by allowing seamless access to data

regardless of their location, the technology used to store and manage it, and the interfaces needed to recover the data. The query federation systems are used to transform one query into a query in many sources and integrated to form a single output.

### 3.4. Schema Evolution and Metadata Management

Schema evolution is a process of updating and changing database structures as the application and data requirements vary. The common changes involve the addition or removal of tables and attributes, field renaming, or data type modification. These updates should be handled with caution so as not to violate linked applications and queries. Metadata management is also important as it stores data concerning schema versions, relationships and transformation rules, which allow compatibility and traceability. An efficient metadata management also provides the ease of migration, eliminates the integration contradictions, and provides the ability to access the data with a consistent set of systems in the field of its evolution. Long-term data integrity, adaptability, and system dependability are guaranteed by the combination of schema evolution and metadata management.

## 4. Nosql Database Context with Big Data Analytics

Big Data, which includes both structured and unstructured information gathered from many sources including social networks, emails, text documents, GPS, sensors, surveillance system data, and more, is growing at a rapid pace in today's world, according to recent innovations [9]. Data that is organised, semi-structured, or unstructured, and comes from digital and non-digital sources is what is known as Big Data. Using Big Data, the data source for effective decision-making, to its full potential requires the right data mining tools, which presents a significant challenge. Currently, big data is plagued by the following common business components. [1]:

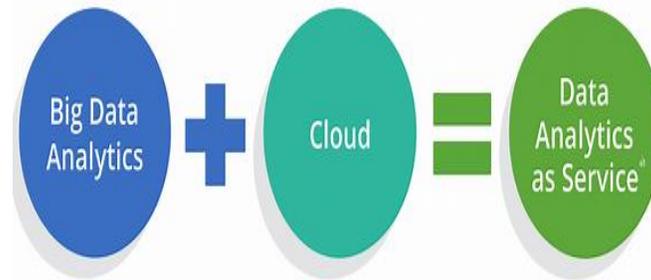
- High data Velocity: Data is quickly and frequently updated from numerous sources and locations.
- Data Variety: Data storage that is semi-structured, unstructured, or organised.
- Volume of data: several datasets with sizes spanning from terabytes to petabytes.
- Complexity information: Data distributed among multiple locations or data hubs.

Big Data analytics is the way that companies look at very big sets of different kinds of data. Businesses may benefit greatly from Big Data Analytics because it improves their data analysis capabilities, which in turn helps them uncover previously unseen patterns, correlations, trends, customer preferences, and market trends [10]. In order to produce effective and rapid choices, big data analytics makes use of clusters of enormous quantities of data. Companies are thinking about jumping on the NoSQL bandwagon since conventional databases can't handle clusters and big data analytics isn't efficient [11].

Here, NoSQL databases triumph over SQL databases due to the former's superior management of data partitioning, processing of dynamic data, and enhanced performance. Facebook, LinkedIn, Amazon, and Google are just a few of the large internet companies that looked into NoSQL as a solution to their problems with vertical large-scale data scalability, optimising data use, and managing ever-increasing amounts of data. While cloud computing and three-tier Internet architecture systems are known for their dependability and dispersed nature, NoSQL databases are a better fit for dynamic, efficient IT systems [12]. This calls for more research into the merits of Big Data analysis in comparison to NoSQL and SQL, and into the notable variations in the performance of NoSQL data models. Within the framework of the current Big Data paradigm, this paper offers the study results.

### 4.1. Big Data Analytics in Cloud

Data analytics in the cloud might not be possible without cloud computing. Any web browser may remotely access the hardware and software that is the foundation of cloud computing [13]. It is common practice to centralise data remotely so that several users may access it, rather than storing it on each user's hard disc. Remote computing Data kept on cloud storage, as opposed to local servers or discs, is typically subject to analytics methods for purposes including tracking social media participation and statistics [14]. Businesses may employ big data analytics to their capacities owing to the cloud's remarkable flexibility. Building dependable and reasonably priced infrastructure may require investing in big data and analytics.



**Figure 2. Big Data Analytics as a Service is a Combination of Big Data Analytics and Cloud**

DaaS is a method for creating an expandable infrastructure that may provide cloud-based computational tools for a variety of use cases and businesses (Figure 2). Technically, the interface comprises all of an analytical solution's capabilities, from data gathering to end-user visualisation, monitoring, and interaction. Analytical Apps and an Analytical Appstore are two novel innovations that are added to the standard features. The optimal setting for big data and business analytics is a hybrid of cloud computing and big data, as the former offers greater flexibility and accessibility [15].

- **Agility:** Data processing and storage methods that were previously used are gradually becoming antiquated. Deploying and running a server, for example, might take weeks, adding significant time and money to the already lengthy process of infrastructure setup. By storing data in the cloud, any infrastructure can get to all of its required services very fast [16]. A good cloud solution can assist organisations ensure that their jobs are still functioning properly.
- **Ease of use:** Cloud infrastructure may expand dynamically to manage increasing volumes of data. Organisations can adjust their storage capacity up or down based on the data they've retrieved.
- **Data processing:** Efficiently managing the massive amount of data is a hurdle. Social networking creates vast volumes of unstructured data in a number of types. When paired with big data technology, cloud computing streamlines and increases accessibility for small, medium, and large businesses.
- **Reduced complexity:** The Big Data processing team's efficiency and accuracy are both enhanced by the automation of each module made possible by cloud computing.
- **Reduce cost:** Businesses may create income by moving procedures and large-scale data processing to the cloud. This is due to the fact that big data analytics implementation on-premise requires companies to own and run large data centres. This minimises IT expenses. When you go to the cloud, this responsibility is passed on to the cloud service provider.

## 5. Hybrid Architectures for Real-Time Big Data Analytics

The complimentary features of SQL and NoSQL systems are used by the hybrid big-data building blocks to provide real-time analytics, scalable storage, and high-velocity processing of massive amounts of data. The suggested NoSQL- SQL hybrid storage platform combines three layers of databases MMDB, RDB, and DFS to obtain access in real time, on-the-fly analysis, and the ability to manage large scale historical data. MMDB allows real time ingestion and querying the latest spatio-temporal data with memory based sliding window storage to reduce latency. RDB layer contains formatted analysis output and triggers to detect events in real-time and publish/subscribe messages to bottom analytics modules. In the meantime, the DFS layer handles historical datasets of great volumes in a distributed and scalable space to support long-term analytics. This hybrid architecture provides the solution of fast in-memory processing, structured relational analysis and distributed storage of historical data to accomplish efficient real-time analytics of big-data and scalable decision-making.

### 5.1. Stream Processing Frameworks Integration

The emergence of many big data streaming frameworks has made real-time, massive-scale stream processing possible. Focussing on the most popular big data stream, this part sheds light on them. processing architectures and provides a comparison analysis of them according to their key characteristics [17].

### 5.2. Apache Spark

Apache Spark, an efficient processing architecture, is a handy analytics tool for conducting successful heterogeneous data analysis. In 2009, UC Berkeley created it. Spark has some advantages over other huge data systems such as Storm and Hadoop mapping. One of Spark's main ideas is RDDs, or resilient distributed datasets. Distributed collections of immutable items are all that constitute an

RDD. Spark is capable of doing both transformations and actions on RDDs. By combining existing RDDs with transformations such as map, filter, union, and join, new RDDs may be generated. RDD calculations yield actions as their end results. Spark streaming is a Spark library that allows for large-scale, fast stream processing of real-time data streams.

### 5.3. Apache Flink

Flink is an open-source system that processes data in either real-time or batch mode. Big data processing and fault tolerance are two of its many capabilities. The programming principles of MapReduce and Flink are identical. Flink offers more advanced operations than MapReduce, such as join, filter, and aggregate. Flink enables real time computation and iterative processing of stream data gathered with the assistance of other tools e.g. Flume and Kafka. It presents a number of APIs that are more abstract and permit the user to initiate spread addition in a clear and simple manner.

### 5.4. Apache Storm

The open-source platform Storm allows for the real-time analysis of massive volumes of organised and unstructured data. Storm is a paradigm for fault-tolerant computing, machine learning, and real-time data processing. The term "directed acyclic graph" (DAG) describes a Storm program. The program's DAG edges represent data transfer. Bolts and spouts are the two types of DAG nodes. The Storm program's spouts (also referred to as entrance points) act as data sources. The bolts reflect the mathematical operations that applied to the dataset. It is crucial to understand that Storm distributes the bolts over several nodes in order to analyse data simultaneously. A master node running Nimbus and a supervisor node on each slave node make up Storm. The Nimbus system oversees and delegate jobs to the slave nodes. It finds another node in the cluster to take over processing when it notices that one of its own has failed. How tasks are carried out is a matter for each supervisor to decide. It is able to commence or avoid the areas based on Nimbus's directions. The Storm cluster is designed to handle topologies that have multiple tasks assigned to them.

### 5.5. Event-Driven Architectures for High-Velocity Data

Event-Driven Architecture (EDA) is now a vital paradigm regarding the processing of high-velocity data streams that are produced continuously by IoT devices, sensors, financial transactions, social media streams, and real-time monitoring systems [18]. Data in EDA is not stored and processed in large batches but rather recorded and processed immediately when an event occurs. This allows systems to respond to real-time changes, which ensures that there is low-latency analytics, quick decision-making, and real-time responsiveness of the system. EDA normally incorporates message brokers like Apache Kafka, RabbitMQ or AWS Kinesis which provide event streaming of high throughput and fault tolerance. Such brokers feed event streams into event processing real-time processing systems, e.g. Apache Flink, Apache Spark streaming, and Apache Storm, to perform continuous compute and complex event processing (CEP). Consequently, event-driven architectures are suitable to support scalable, loosely-coupled systems with the capacity to support millions of events per second, suitable in mission-critical applications in smart cities, healthcare, autonomous systems, and real-time financial analytics.

### 5.6. Use of AI/ML for Performance Prediction and Scaling

ML and AI are essential components of the current cloud and big-data systems as they allow predicting the performance accurately and scaling the system in an intelligent manner. Conventional rule-based scaling or threshold scaling is not able to effectively cope with unforeseeable and fast evolving workloads [19]. The models of AI/ML use present and past data on CPU, memory, network, and user requests to predict the future needs of the resources. Systems can preempt resource scaling rather than responding to resource demands by forecasting the upsurges or declines in workload. Most popular deep learning techniques include LSTM, Bi-LSTM, ARIMA, CNN, and reinforcement learning that enhance accuracy of prediction and minimize response time. ML-based auto-scaling can reduce the latency, cost, and performance degradation and enhance the reliability of real-time applications such as IoT analytics, streaming systems, and large-scale cloud platforms. By incorporating AI-driven predictive models into orchestration systems, including Kubernetes or OpenStack, it is possible to make systems more scalable and resource-optimizing, as well as to achieve automated decision-making and be more sustainable and efficient.

## 6. Literature Review

The literature reviewed discusses SQL-to-NoSQL transformation plans, scalable Big Data systems, distributed database systems, and the attributes of a NoSQL data store. It outlines architectural innovations, comparative assessments, and major issues and focuses on the areas of performance, completeness and optimization issues that need research and enhancement of practical implementation.

Schreiner, Duarte and Mello (2019) provides a comparison of different approaches utilising an architectural taxonomy that classifies them based on their system architectures. When relational database systems need to transfer data to NoSQL databases, which are often hosted in the cloud, wrapping is a helpful technique. The fact that most methods only deal with a subset of SQL procedures or certain NoSQL databases further adds to my belief that this area of study faces substantial obstacles. Finding or building a relational-to-NoSQL data wrapping solution can be aided by this review, which aims to advance the state of the art in this area of research [20].

Alotaibi and Pardede (2019) Create RDB Schema change rules for different NoSQL database layouts, such as document-based, column-based, and graph-based databases. All possible interactions with data stored in a database are considered while formulating the rules. A case study utilising three NoSQL databases MongoDB, Cassandra, and Neo4j proved the hypotheses. To show how accurate the transformation results are, several queries are run in different databases. Additionally, the completeness of the transformation rules is examined in reference to earlier research [21].

Ezéchiel, Kant and Agarwal, (2019) offers a thorough analysis of distributed database systems using the three distribution techniques of data replication, data allocation, and data fragmentation. Some of the obstacles faced in creating and executing these strategies were discussed. Data fragmentation hinders join optimisation as a query must combine more than one fragment stored across various locations. This has resulted in a speedy response. Heuristic techniques have been studied to cope with this obstacle, which is known as an NP-hard problem [22].

Mazumdar et al., (2019) present a cutting-edge review of cloud-centric Big Data deployment and storage approaches. To effectively support Big Data management, it is necessary to draw attention to the true relationship between the two. attention is on managerial elements which are considered in the prism of non-functional qualities. The comprehensive analysis of certain Big Data management solutions is a satisfying last step for readers who want to know how to satisfy their non-functional application needs. Aside from that, issues are handled, drawing attention to the current shortcomings in Big Data management and outlining the strategy that has to be altered soon [23].

Sarnovsky, Bednar and Smatana (2018) presents the blueprint for a process industry-specific Big Data platform that integrates data from several disciplines. Making an extensible analytics platform that could take in data from different sectors and process it was the main objective. Connecting to the plant's existing environment and using the data acquired to construct predictive functions that enhance production processes should be possible with this platform. In order to test the models, the analytical platform has a simulation environment, and in order to build these features, it contains a development environment [24].

Ashgari and Zarrabi, (2016) presents an overview on NoSQL and its many data storage options. Large amounts of data are saved in NoSQL (Not Only SQL) databases. NoSQL databases are distributed, open-source, not relational, and horizontally scalable. In this article important concepts like ACID, BASE and CAP theorem presented and offer a comparison between ACID and BASE properties. Along with Key-Value, Column Family, Document, and Graph databases, the CAP theorem is employed to characterise the attributes of various NoSQL data store types [25].

The literature review Table I identifies various scholarly contributions to relational-to-NoSQL transformation, distributed and big data frameworks, and NoSQL database features, proposing methodologies and comparative findings and other important observations and future research agendas as part of the scalability, completeness, and performance dilemma.

**Table 1. Recent Studies on Hybrid Nosql-SQL Database Architectures for Real-Time Big Data Analytics**

Author	Proposed Work	Results	Key Findings	Limitations & Future Work
chreiner, Duarte and Mello (2019)	Comparative analysis of relational-to-NoSQL database wrapping approaches based on architectural classification.	Identified strengths and weaknesses of different wrapping models.	Wrapping enables seamless migration from SQL to NoSQL for cloud-based applications but lacks full SQL operation support.	Limited support for complex SQL operations; Future work includes designing more comprehensive wrapping frameworks.

Alotaibi and Pardede (2019)	Proposed transformation rules from RDB schema to document, column, and graph-based NoSQL databases using case study on MongoDB, Cassandra, and Neo4j.	Demonstrated correct transformation through query execution; validated rule completeness vs. existing methods.	Schema transformation enables systematic mapping from relational to NoSQL structures.	Limited experimental scenarios; needs evaluation across larger datasets and enterprise-level systems.
Ezéchiél, Kant and Agarwal (2019)	An exhaustive analysis of distributed databases, with an emphasis on methods for data allocation, replication, and fragmentation.	Highlighted efficiency issues and NP-hard join optimization problem.	Database distribution strategies impact response time and workload balancing.	Optimization challenges remain; more intelligent heuristics and ML-assisted algorithms needed.
Mazumdar et al. (2019)	Evaluation of Big Data storage and placement approaches focused on the cloud, with an eye towards scalability in administration and an emphasis on non-functional attributes.	Highlighted correlation between placement strategies and Big Data scalability.	Identified gaps in real-time and efficient cloud management.	Need better methods for the Big Data lifecycle, analytics in real-time, and where to put storage to save money.
Sarnovsky, Bednar and Smatana (2018)	Designed a scalable Big Data platform for cross-sector industrial analytics to support predictive functions and real-time decision support.	Created a functional analytical and simulation environment integrated with plant systems.	Demonstrated capability to integrate multiple domains for industrial Big Data analytics.	Requires advanced real-time automation and wider domain deployment; scalability under extreme loads not fully tested.
Asghari and Zarrabi, (2016)	Survey of NoSQL data stores including ACID vs BASE and CAP-based classification of NoSQL architectures.	Provided comparison of four NoSQL categories and clarified performance trade-offs.	NoSQL supports scalability and flexible schema but compromises strong consistency.	Hybrid SQL-NoSQL integration and consistency trade-offs require more research.

### 6.1. Research Gap

Despite the currently available literature on relational-to-NoSQL transformation, distributed database strategies, and Big Data storage methodologies, there are still substantial gaps in terms of developing fully integrated, scalable and consistent hybrid SQL-NoSQL systems. Existing transformation models only accommodate partial SQLs and do not provide end to end mapping of complex schemas. Distributed databases continue to have problems with the NP-hard optimization of joins, inefficient fragmentation, and poor real-time performance when faced with large scale workloads. Big Data solutions in the clouds need to have a better data location mechanism, effective cost control, and more automation of real-time analytics. Besides, consistency control between SQL systems with the ACID (and BASE-oriented NoSQL) stores is not deeply investigated, especially in high-velocity systems. Important AI/ML-based orchestration is also required to perform auto-scaling, minimize latency, and streamline hybrid designs. In general, there is no coherent, well-built system that can be found throughout the literature that can be incorporated into the real-time ingestion, scalable storage, low-latency analytics, and intelligent resource management.

## 7. Conclusion and future study

The rapid rise in data volume, velocity, and variety has exposed classic RDBMSs' limitations in satisfying the analytical and scalability needs of modern applications. The current study highlights the necessity for hybrid SQL-NoSQL systems to address the expanding needs of real-time Big Data analytics, where the scalability, low latency, and heterogeneous data processing requirements cannot be satisfied by ordinary relational databases. The proposed framework allows an efficient, scalable and fault tolerant data management in modern cloud environments by integrating MMDB as a real-time ingestion framework, RDB as structured on-the-fly analytics framework, and DFS as a high volume historical storage framework. The interoperability, streaming system and AI/ML-based scaling discussion further substantiates the relevance of integrated architectures that have the ability to accommodate unlimited data expansion and dynamism workloads. The study is limited to some extent as it has not been experimentally validated on large scale data and as to full coverage of ACID-BASE consistency issues. The effects of schema evolution, query translation and multi-source integration

are also not analyzed completely. The future work should produce the automated schema transformations, smart query optimization, and ML-based consistency management. There is a need to expand benchmarking, better event-driven architectures and better auto-scaling strategies, as well as to generalize the hybrid model to edge and IoT environment.

## References

- [1] S. Venkatraman, S. Kaspi, Kiran Fahd, and R. Venkatraman, "SQL Versus NoSQL Movement with Big Data Analytics," *Int. J. Inf. Technol. Comput. Sci.*, vol. 8, no. 12, pp. 59–66, Dec. 2016, doi: 10.5815/ijitcs.2016.12.07.
- [2] S. A. Hossain, "NoSQL Database : New Era of Databases for Big data Analytics - Classification , Characteristics and Comparison," vol. 6, no. 4, pp. 1–14, 2013.
- [3] C. Băzăr, "The Transition from RDBMS to NoSQL. A Comparative Analysis of Three Popular Non-Relational Solutions: Cassandra, MongoDB and Couchbase," *Database Syst. Journa*, vol. V, no. 2, pp. 49–59, 2014.
- [4] V. K. Tambi, "Analysis Of Sql And Nosql Database Management," vol. 2, no. 3, pp. 99–113, 2015.
- [5] B. R. Cherukuri, "Future of cloud computing: Innovations in multi-cloud and hybrid architectures," *World J. Adv. Res. Rev.*, vol. 1, no. 1, pp. 068–081, 2019.
- [6] C. Wu *et al.*, "A NoSQL-SQL Hybrid Organization and Management Approach for Real-Time Geospatial Data: A Case Study of Public Security Video Surveillance," *Int. J. Geo-Information*, vol. 6, p. 21, 2017, doi: 10.3390/ijgi6010021.
- [7] M. N. Mami, D. Graux, H. Thakkar, S. Scerri, and S. Auer, "The Query Translation Landscape : a Survey," no. 1, 2019.
- [8] N. Fikri, M. Rida, N. Abghour, K. Moussaid, and A. El Omri, "An adaptive and real-time based architecture for financial data integration," *J. Big Data*, vol. 6, no. 1, p. 97, 2019, doi: 10.1186/s40537-019-0260-x.
- [9] A. Pothuganti, "Big Data Analytics: Hadoop-Map Reduce & NoSQL Databases," *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 1, 2015.
- [10] V. M. L. G. Nerella, "Automated Cross-Platform Database Migration And High Availability Implementation," *Turkish J. Comput. Math. Educ.*, vol. 9, no. 2, pp. 823–835, Jul. 2018, doi: 10.61841/turcomat.v9i2.15284.
- [11] W. Ali, M. U. Shafique, M. A. Majeed, and A. Raza, "Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics," *Asian J. Res. Comput. Sci.*, pp. 1–10, Oct. 2019, doi: 10.9734/ajrcos/2019/v4i230108.
- [12] Y.-L. Choi, W.-S. Jeon, and S.-H. Yoon, "Improving Database System Performance by Applying NoSQL," *J. Inf. Process. Syst.*, vol. 10, no. 3, pp. 355–364, Sep. 2014, doi: 10.3745/JIPS.04.0006.
- [13] A. Kobusińska, C. Leung, C.-H. Hsu, R. S., and V. Chang, "Emerging trends, issues and challenges in Internet of Things, Big Data and cloud computing," *Futur. Gener. Comput. Syst.*, vol. 87, pp. 416–419, Oct. 2018, doi: 10.1016/j.future.2018.05.021.
- [14] A. Kushwaha, P. Pathak, and S. Gupta, "Review of Optimize Load Balancing Algorithms in Cloud.," *Int. J. Distrib. Cloud Comput.*, vol. 4, no. 2, p. 1, 2016.
- [15] B. M. Balachandran and S. Prasad, "Challenges and Benefits of Deploying Big Data Analytics in the Cloud for Business Intelligence," *Procedia Comput. Sci.*, vol. 112, pp. 1112–1122, 2017, doi: 10.1016/j.procs.2017.08.138.
- [16] S. Achouche and U. B. Yalamanç, "Method, apparatus, and computer-readable medium for performing a data exchange on a data exchange framework," 2019.
- [17] W. Inoubli, S. Aridhi, H. Mezni, and M. Maddouri, "A Comparative Study on Streaming Frameworks for Big Data," pp. 17–24, 2018.
- [18] Q. Zhou, Y. Simmhan, and V. Prasanna, "Knowledge-infused and consistent Complex Event Processing over real-time and persistent streams," *Futur. Gener. Comput. Syst.*, vol. 76, pp. 391–406, Nov. 2017, doi: 10.1016/j.future.2016.10.030.
- [19] S. Garg, "AI/ML Driven Proactive Performance Monitoring, Resource Allocation and Effective Cost Management in SaaS Operations," *Int. J. Core Eng. Manag.*, vol. 6, no. 6, pp. 263–273, 2019.
- [20] G. A. Schreiner, D. Duarte, and R. dos S. Mello, "When Relational-Based Applications Go to NoSQL Databases: A Survey," *Information*, vol. 10, no. 7, 2019, doi: 10.3390/info10070241.
- [21] O. Alotaibi and E. Pardede, "Transformation of Schema from Relational Database (RDB) to NoSQL Databases," *Data*, vol. 4, no. 4, p. 148, Nov. 2019, doi: 10.3390/data4040148.
- [22] K. Ezéchiél, S. Kant, and D. Agarwal, "A systematic review on Distributed Databases Systems and their techniques," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 1, 2019.
- [23] S. Mazumdar, D. Seybold, K. Kritikos, and Y. Verginadis, "A survey on data storage and placement methodologies for Cloud-Big Data ecosystem," *J. Big Data*, vol. 6, no. 1, p. 15, 2019, doi: 10.1186/s40537-019-0178-3.
- [24] M. Sarnovsky, P. Bednar, and M. Smatana, "Big Data Processing and Analytics Platform Architecture for Process Industry Factories," *Big Data Cogn. Comput.*, vol. 2, no. 1, 2018, doi: 10.3390/bdcc2010003.
- [25] P. Asghari and H. Zarrabi, "A Survey on NoSQL and its Terminology," vol. 5, no. 12, pp. 405–410, 2016, doi: 10.17148/IJARCC.2016.51293.
- [26] Mamidala, J. V., Enokkaren, S. J., Attipalli, A., Bitkuri, V., Kendyala, R., & Kurma, J. (2023). Machine Learning Models Powered by Big Data for Health Insurance Expense Forecasting. *International Research Journal of Economics and Management Studies IRJEMS*, 2(1).
- [27] Nadella, V. M. (2023). Zero Trust Architecture for Telecom Operations. *International Journal of Emerging Research in Engineering and Technology*, 4(3), 115-129.

- [28] Bitkuri, V., Kendyala, R., Kurma, J., Enokkaren, S. J., & Mamidala, J. V. (2023). Forecasting Stock Price Movements With Deep Learning Models for time Series Data Analysis. *Journal of Artificial Intelligence & Cloud Computing*. SRC/JAICC-531. DOI: [doi.org/10.47363/JAICC/2023\(2\),489](https://doi.org/10.47363/JAICC/2023(2),489), 2-9.
- [29] Nadella, V. M. (2023). Anomaly Detection and Fault Prediction using ML in Telecom Operations. *International Journal of Emerging Trends in Computer Science and Information Technology*, 4(3), 134-143.
- [30] Kosaraju, P., & Nadella, V. M. (2022). Security and Privacy in IoT Ecosystems. *Universal Library of Engineering Technology*, (Issue).
- [31] Singh, A. A. S. S., Mania, V., Kothamaram, R. R., Rajendran, D., Namburi, V. D. N., & Tamilmani, V. (2023). Exploration of Java-Based Big Data Frameworks: Architecture, Challenges, and Opportunities. *Journal of Artificial Intelligence & Cloud Computing*, 2(4), 1-8.
- [32] Routhu, K. K. (2023). AI-driven succession planning in Oracle HCM Cloud: Building resilient leadership pipelines through predictive analytics. *International Journal of Science, Engineering and Technology*, 11(5).
- [33] Tamilmani, V., Namburi, V. D., Singh Singh, A. A., Maniar, V., Kothamaram, R. R., & Rajendran, D. (2023). Real-Time Identification of Phishing Websites Using Advanced Machine Learning Methods. Available at SSRN 5837142.
- [34] Routhu, K. K. (2023). AI-driven succession planning in Oracle HCM Cloud: Building resilient leadership pipelines through predictive analytics. *International Journal of Science, Engineering and Technology*, 11(5). <https://doi.org/10.5281/zenodo.17292018>
- [35] From Fragmentation to Focus: The Benefits of Centralizing Procurement. (2023). *International Journal of Research and Applied Innovations*, 6(6), 9820-9833. <https://doi.org/10.15662/>
- [36] Routhu, K. K. (2023). Embedding fairness into the digital enterprise, data driven DEI strategies with Oracle HCM Analytics. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(8), 266-274.
- [37] Routhu, K. K. (2023). AI-driven skills forecasting in Oracle HCM Cloud: From static competencies to predictive workforce design. *International Journal of Science, Engineering and Technology*, 11(1).
- [38] Padur, S. K. R. (2023). AI-Augmented Enterprise ERP Modernization: Zero-Downtime Strategies for Oracle E-Business Suite R12. 2 and Beyond. Available at SSRN 5605510.
- [39] Routhu, K. K. (2022). From Case Management to Conversational HR: Redefining Help Desks with Oracle's AI and NLP Framework. *International Journal of Science, Engineering and Technology*, 10(6).
- [40] Vattikonda, N., Gupta, A. K., Polu, A. R., Narra, B., Buddula, D. V. K. R., & Patchipulusu, H. H. S. (2022). Blockchain Technology in Supply Chain and Logistics: A Comprehensive Review of Applications, Challenges, and Innovations. *International Journal of Emerging Trends in Computer Science and Information Technology*, 3(3), 72-80.
- [41] Attipalli, A., BITKURI, V., Mamidala, J. V., Kendyala, R., & KURMA, J. (2022). Empowering Cloud Security with Artificial Intelligence: Detecting Threats Using Advanced Machine learning Technologies. Available at SSRN 5741263.
- [42] Padur, S. K. R. (2022). Intelligent resource management: AI methods for predictive workload forecasting in cloud data centers. *J. Artif. Intell. Mach. Learn. & Data Sci*, 1(1), 2936-2941.
- [43] Nadella, V. M. (2022). Digital Twins for Predictive Network Management and System Simulation. *International Journal of AI, BigData, Computational and Management Studies*, 3(3), 100-111.
- [44] Routhu, K. K. (2022). From RFID to Geofencing: IoT-Enabled Smart Time Tracking in Oracle HCM Cloud. *International Journal of Science, Engineering and Technology*, 10(4).
- [45] Nadella, V. (2019). Extracting road traffic data through video analysis using automatic camera calibration and deep neural networks.
- [46] Polam, R. M., Kamarthapu, B., Kakani, A. B., Nandiraju, S. K. K., Chundru, S. K., & Vangala, S. R. (2022). Data Security in Cloud Computing: Encryption, Zero Trust, and Homomorphic Encryption. *International Journal of Emerging Trends in Computer Science and Information Technology*, 3(4), 31-41.
- [47] Padur, S. K. R. (2022). AI augmented platform engineering, transforming developer experience through intelligent automation and self optimizing internal platforms. *International Journal of Science, Engineering and Technology*, 10(5), 10-5281.
- [48] Kosaraju, P. , & Nadella, V. M. (2021). Quality of Experience (QoE) and Network Performance Modelling for Multimedia Traffic. *Journal of Artificial Intelligence and Big Data*, 1(1), 1-13. <https://doi.org/10.31586/jaibd.2021.1358>