International Journal of Core Engineering & Management
Volume-7, Issue-02, 2022 ISSN No: 2348-9510

IMPORTANCE OF NOSQL DATABASES: BUSINESS STRATEGIES WITH ADMINISTRATION TACTICS

Balakrishna Boddu
balakrishnasvkbs@gmail.com
Sr. Database Administrator

Abstract

Nowadays, the quantity and size of data have grown tremendously. It is essential to manage SQL or NoSQL data, particularly when the data comes in the form of files, images or unstructured, or semi-structured. This paper demonstrates the necessity for a system of specialized NoSQL database administration methods that can meet the demands of real-time applications. Topics covered included data modeling, performance optimization, scalability, high availability, security, and other measures to support that level which aims at development and provides on an ongoing basis continuously monitoring. By utilizing these methods, businesses can make sure their data management is efficient reliable, and secure, giving rise to the performance and reaction abilities of real-time applications. This talk gets confirmation through several practical examples from actual deployment: in e-commerce, for instance, financial services, IoT, and social media. These show how effective management by a NoSQL system will bring about a transformative change in each area.

Keywords: Bigdata, Redshift, Big Query, Snowflake, cloud computing, document databases, MongoDB, NoSQL

I. INTRODUCTION

Since NoSQL Databases Were Introduced onwards, they have been challenging the relational structures and continuous database management techniques based on traditional SQL databases. Over the years, the hype that NoSQL is meant to replace relational databases has become quite strong. But NoSQL should be thought of as a combination of non-relational databases with relational ones, the goal of this approach is to attempt to take full advantage of all available technologies in order not just to achieve but to balance the requirements for both performance, scalability, and data consistency on one hand but schema flexibility and robustness on the other. Traditional relational databases were not designed to cope with the scalability and flexibility challenges facing modern applications. Older databases were not designed to take advantage of the inexpensive storage and excess processing power available today.

The growing amount of research on NoSQL technologies includes studies on their unique features, benefits, drawbacks, and how this has affected their adoption. This paper summarizes this existing research and suggests potential areas for future study.

The rapid growth of data volumes and the increasing complexity of modern applications have led to a paradigm shift in data management. Traditional relational databases, while valuable for structured data, often struggle to keep pace with the diverse and dynamic nature of data generated

today. NoSQL databases, designed to handle unstructured, semi-structured, and highly scalable data, have emerged as a compelling alternative for businesses seeking to leverage their data effectively. This paper explores the critical role of NoSQL databases in modern business strategies, highlighting the key benefits they offer and the essential administration tactics required for successful implementation. We will delve into the various types of NoSQL databases, their use cases, and the challenges and considerations that must be addressed by organizations adopting this technology.

II. REVIEW OF LITERATURE

NoSQL databases are relatively new, so much of the research about them is still developing. Terms like NoSQL, relational databases, non-relational databases, semi-structured databases, unstructured databases, document databases, big data, business intelligence, data warehouse, OLAP, OLTP, index optimization, MongoDB, database consistency, eventual consistency, database scalability, and NoSQL adoption.

Research shows that NoSQL databases are more flexible and scalable than traditional SQL databases, but they lack some features that SQL databases have. A lot of the research about NoSQL focuses on classifying different types of NoSQL databases, comparing their advantages and disadvantages, finding ways to address their shortcomings (especially in terms of consistency, querying, and working with other systems), and understanding why organizations are adopting NoSQL technologies.

III. TYPES OF NOSQL STRUCTURES

NoSQL databases come in many different types, each designed for specific purposes. There are many vendors and systems in each category of NoSQL databases.

Document databases store data in key-value pairs with formats like XML, JSON, and BSON. Values are stored in these semi-structured data formats, and each document can have hundreds of attributes with different data types. Examples of document databases include Couch DB, AWS Document DB, and MongoDB.

Key-value stores store alphanumeric keys with associated values in separate hash tables. They are good for quickly and easily retrieving values. Examples of key-value stores include MemcacheDB and Dynamo DB.

Wide-column stores are useful for storing large amounts of data in a distributed way. These systems have time-stamping functions that help manage versioned data. They can also be used for predictive analytics.

Graph databases use structured, interconnected graphs to connect data instead of tables. Graph databases are best for analysing the relationships between data, not the data itself.

International Journal of Core Engineering & Management
Volume-7, Issue-02, 2022 ISSN No: 2348-9510

	Structured Data	Unstructured data
Characteristics	Pre-defined data models	No Pre-defined-data model
	Usually text only	Maybe text, image, sound, video or Other formats
	Easy to search	Difficult to search
Resides in	Relational databases	Applications
	Data warehouses	NoSQL databases
		Data lakes
		Data warehouses
Generatedby	Humans or machines	Humans or machines
Typical applications	Airline reservation systems	Word processing
	Inventory control	Presentation software
	ERP systems	Email clients
	CRM systems	Tools for viewing or editing media
Examples	Dates	Text files
	Phone numbers	Reports
	Social security numbers	Email messages
	Credit card numbers	Audio files
	Customer names	Video files
	Addresses	Images
	Product names and numbers	Surveillance imagery
	Transaction Information	

	AWS Document DB	Mongo DB	Redshift	Hadoop	Redis	Memcache DB	Dynamo DB
Storage Structure	JSON	Binary JSON	Columner	Map-R	Cache	cache	JSON
Data Types	Number	string	SMALLINT	Int	String	Int	bool
	String	Integer	INTEGER	long	Hash	String	byte
	Boolean	double	BIGINT	float	List		char
	Array	Boolean	DECIMAL	Double	Set		DateTime
	Value	null	REAL	Boolean	Sorted set		decimal
	Object	Array	DOUBLE PRECISION	String	Stream		double
	WhiteSpace	Object	CHAR	Map	Bitmap		float
	Null	Object id	VARCHAR	struct	Bitfield		Guid
		Undefined	DATE	union	Geospatial		Int16
		Binary Data	TIME	Array			
		Min & Max key	TIMETZ				
		Symbol	TIMESTAMP				
		Regular Expression	TIMESTAMPTZ				
		JavaScript	INTERVAL YEAR TO MONTH				
		JavaScript with Scope	INTERVAL DAY TO SECOND				
		Timestamp	BOOLEAN				
		Decimal	HLLSKETCH				
			SUPER				
			VARBYTE				
			GEOMETRY				
			GEOGRAPHY				

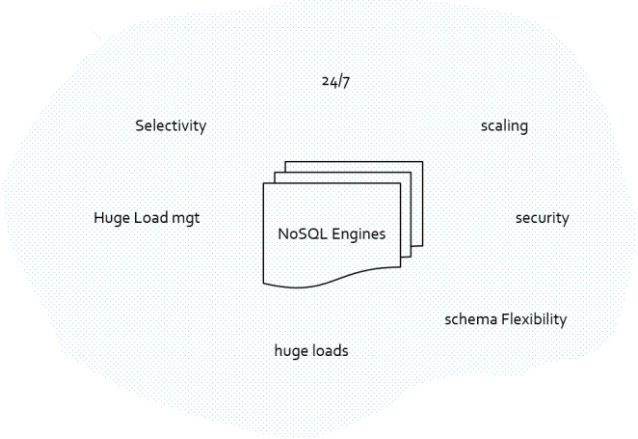
Diagram: Data types For NoSQL Databases

IV. CHARACTERISTICS OF NOSQL ADMINISTRATION

NoSQL databases have gained significant improvement in recent years, due to their scalability, flexibility, and performance advantages. As their adoption continues to grow, so does the need for effective administration techniques. Managing NoSQL Databases as a Database administrator is

International Journal of Core Engineering & Management
Volume-7, Issue-02, 2022 ISSN No: 2348-9510

challenging due to lake of knowledge and there is no source of reference in the market too.



1. Schema models:

Schema and No Schema: NoSQL databases often adopt a schema-less or schema-on-read approach, allowing for dynamic data structures and easier schema design.

Document-Oriented, Key-Value, Graph, or Wide-Column: Each NoSQL database type has its own unique data model, requiring specific administration techniques.

Data Modelling Best Practices: Understanding the strengths and limitations of each data model is crucial for designing efficient and scalable NoSQL applications.

2. Always-on Model

Sharding: NoSQL databases often employ sharding to distribute data across multiple nodes, ensuring scalability and performance.

Replication: Replication strategies are essential for data redundancy and fault tolerance.

Consistency Models: Understanding eventual consistency, strong consistency, and causal consistency is vital for managing data consistency in distributed systems.

3. High Availability

Horizontal Scaling: NoSQL databases typically scale horizontally, allowing for easy addition of nodes to handle increasing workloads.

Performance Optimization: Techniques like indexing, caching, and query optimization are crucial for achieving optimal performance.

Workload Analysis: Understanding the workload characteristics of NoSQL applications is essential for effective performance tuning.

4. Data Management Challenges

Data Consistency: Ensuring data consistency in distributed systems can be challenging, especially with eventual consistency models.

Schema Evolution: Managing schema changes in NoSQL databases can require careful planning and testing.

Query Optimization: Optimizing queries in NoSQL databases can be different from traditional SQL databases, requiring specific techniques.

International Journal of Core Engineering & Management
Volume-7, Issue-02, 2022 ISSN No: 2348-9510

5. Administration Tools and Techniques

Monitoring and Alerting: Effective monitoring tools are essential for identifying performance issues and anomalies.

Backup and Recovery: Implementing robust backup and recovery strategies is crucial for data protection and disaster recovery.

Security: Ensuring data security by implementing access controls, encryption, and regular security audits.

Automation: Automating administrative tasks can improve efficiency and reduce human error.

6. Latest Trends:

Server less NoSQL: Exploring the benefits and challenges of server less NoSQL databases.

Hybrid Data Management: Combining NoSQL and relational databases for specific use cases.

AI and Machine Learning: Leveraging AI and machine learning for tasks like data modelling, query optimization, and anomaly detection.

7. Tools:

There are a good number of tools that are helping DBA to monitor and take proactive action when any issues happen from the servers.

Like site24/7, Data dog, Cloud watch Alarms for Redshift, Dynamo DB, Document DB, and others.

Data dog is the market leader in getting a good number of metrics and it will send an alert to administrators for proactive monitoring.

Diagram: Dashboard from Datadog

8. OLTP vs OLAP

OLTP handles smaller, more frequent transactions, while OLAP deals with large volumes of data.

OLTP queries are simple and short, focusing on data manipulation. OLAP queries are complex and long, focusing on data analysis. NoSQL Databases like MongoDB can handle PLTP workload and Redshift, and Cassandra can handle terabytes of data in the real world.

V. RESEARCH AREAS FOR NOSQL ADMINISTRATION

1. Hybrid NoSQL-Relational Database Management

Integration Strategies: Investigate effective strategies for integrating NoSQL and relational databases to leverage the strengths of both.
Data Migration and Synchronization: Develop efficient methods for migrating data between NoSQL and relational databases and ensuring data consistency.
Query Optimization: Explore techniques for optimizing queries that span multiple database types.

2. AI-Driven NoSQL Administration:

Anomaly Detection: Apply AI and machine learning algorithms to automatically detect anomalies and performance issues in NoSQL databases.
Query Optimization: Use AI to optimize query execution plans and improve performance.
Predictive Maintenance: Leverage AI to predict potential issues and proactively address them before they impact performance.

3. Server less NoSQL Database Performance and Scalability

Performance Benchmarking: Conduct in-depth performance benchmarks of server less NoSQL databases to evaluate their scalability and efficiency.
Cost Optimization: Explore strategies for optimizing costs in server less NoSQL environments, considering factors like usage-based pricing and resource allocation.
Best Practices: Identify best practices for designing and managing server less NoSQL applications.

4. NoSQL Database Security Best Practices

Data Encryption: Investigate effective encryption techniques for securing data in NoSQL databases.
Access Controls: Develop robust access control mechanisms to protect sensitive data.
Vulnerability Assessment: Identify and address potential security vulnerabilities in NoSQL databases.

5. NoSQL Adoption and Migration Strategies

Case Studies: Analyse successful NoSQL adoption case studies to identify common challenges and best practices.
Migration Planning: Develop guidelines for migrating from relational databases to NoSQL databases, considering factors like data volume, complexity, and performance requirements.
Cost-Benefit Analysis: Evaluate the potential costs and benefits of adopting NoSQL databases for specific use cases.

VI. Considering NoSQL Databases Real Time Learning

Many of them thinking about NoSQL databases Not sure table for some of the fast-facing Applications let's discuss that.

Point 1: Relationship data is best suited for relational databases A common misconception is that NoSQL databases or non-relational databases don't store relationship data well. NoSQL databases can store information about how things are connected, just like relational databases. However, they store this information in a different way. NoSQL databases often group related data together in a single structure, making it easier to work with compared to relational databases, which split data across multiple tables.

Point 2: NoSQL databases don't support ACID transactions Many people think that NoSQL databases can't handle complex transactions like relational databases. But this isn't always true. Some NoSQL databases, like MongoDB, can handle these kinds of transactions.
NoSQL databases often store related information together in a single structure, which can simplify things. For example, imagine storing information about a person and their hobbies. In a relational database, you might need to update multiple tables to keep the information consistent. In a NoSQL database, you could update all the information in a single document, making it easier to manage

VII. FUTURE PLANS OF NOSQL RESEARCH

While the body of research concerning NoSQL database technologies continues to grow, specific opportunities exist for improving both academic and practical understanding of NoSQL technologies, including:

- Comparative analyses of NoSQL technologies, by NoSQL classification and across specific vendors
- Further comparisons of NoSQL and SQL databases to assess their performance and scalability in various scenarios
- Proposals for establishing consistent best practices in NoSQL data/document models
- Study of security management capabilities, issues, and opportunities in NoSQL technologies
- Case studies adoption of NoSQL databases in specific use cases and business domains

Additional studies in each of these areas could provide IT organizations with the ability to make a more informed choice in their selection of database technologies. a wide range of sources on database management were examined, with only a small subset specifically addressing NoSQL (the central focus of this paper). Future revisions of this paper may expand upon the findings from the broader review. However, the initial content analysis of sixty-two articles, which is included in Appendix A, has been conducted. This analysis aimed to identify the key themes of each article and resulted in the development of ten categories to classify their primary areas of focus

VIII. CONCLUSION

1. NoSQL databases provide a variety of benefits, including flexible data models, horizontal scaling, lightning-fast queries, and ease of use for developers.
2. NoSQL databases come in a variety of types, including document stores, key-values databases, wide-column stores, graph databases, and multi-model databases.

REFERENCES

1. "Designing Data-Intensive Applications" by Martin Kleppmann: https://www.amazon.com/designing-data-intensive applications/s?k=designing+data+intensive+applications
2. "NoSQL Distilled" by Patrick Nielsen: https://www.amazon.com/NoSQL-Distilled-Emerging-Polyglot-Persistence/dp/0321826620
3. "Seven Databases in Seven Weeks" by Eric Freeman and Elisabeth Robson: https://www.oreilly.com/library/view/seven-databases-in/9781941222829/
4. Dataversity Blog: https://www.dataversity.net/category/blogs/ (Covers a range of data management topics, including NoSQL)
5. NoSQL Now! Blog: https://www.mongodb.com/resources/basics/databases/nosql-explained (Focuses specifically on NoSQL databases)
6. Martin Kleppmann's Blog: https://martin.kleppmann.com/ (Written by the author of "Designing Data-Intensive Applications")
7. Choosing the right NoSQL database for the job: a quality attribute evaluation - This paper evaluates various NoSQL databases based on quality attributes(https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0025-0)
8. Firestore: The NoSQL Server less Database for the Application Developer - This paper discusses Google's Fire store, a NoSQL server less database. (https://research.google/pubs/firestore-the-nosql-serverless-database-for-the-application-developer/)
9. A systematic analysis of trending NoSQL database tools and techniques - This paper provides a comprehensive survey and comparative study of trending NoSQL databases(https://pubs.aip.org/aip/acp/article-abstract/2782/1/020091/2896655/A-systematic-analysis-of-trending-NOSQL-database?redirectedFrom=fulltext)
10. NoSQL Database: New Era of Databases for Big Data Analytics - This report helps users understand the strengths and weaknesses of various NoSQL database approaches(https://arxiv.org/abs/1307.0191)