Big Data and NoSQL Databases Architecture: A Review

Moko, Anasuodei
Department of Computer Science and Informatics,
Federal University Otuoke,
Bayelsa State, Nigeria.

Asagba, Prince Oghenekaro
Department of Computer Science,
University of Port Harcourt,
Rivers State, Nigeria.

Abstract
Data in recent times has become bigger and bigger to handle increasing exceptionally in volume, causing specifically unstructured data to become difficult to process and manage, because structured database experiences difficulties when handling unstructured data due to its size, which is termed Big Data. Hence to make sense of Big data, new architectures and methods are needed, this paper reviews Big Data and its features, the available four NoSQL architecture for handling Big Data, strengths, weaknesses, and types that are available are discussed.

Keywords: *Big Data,Unstructured, NoSQL, Database, Structured*

I. Introduction

One major technological challenge being faced by the world is in data management and storage, in that millions of data over recent times is being produced at intermissions of less than nanosecond. Therefore, handling vast quantity of data is of significant challenge and consequently with the rise in population growth there is need of the state-of-the-art data management and collection technology.

(Sultana, et al, 2017) emphasized that the need for rapid data processing and generation in recent times has resulted in the fact that beyond than 2.6 trillion of data is being produced daily. They further predicted that there will be in future a more exponential increase of data usage and generation in future bearing in mind the way it's being used today. Stating the obvious that data management and storage in the era of big data where conventional software are being pushed to the limits by enormous amount of data, creating the need for significant change investing, acquiring and storing data for future use, thereby making different organizations put in conscious effort to secure and keep every potential piece of data. Though the data is typically unstructured it can be generated from a broad variety of sources, such as post on social media, multimedia content with automated archives. Emails, search engine queries, content management document repositories, sensor data of varying sorts, stock exchange, satellite images, monitoring systems and application of e-health.etc.

It is sometimes interpreted as 'Not Only SQL,' so that it can express the fact that other technologies are used in mass-distributed web applications, besides relational data technologies. Most importantly, NoSQL technologies are required if high availability is required by the web service. It is a widely distributed storage architecture, that has basic useful database management system structure. The actual data of key values is stored in pairs, columns or families of columns, documents and graphs. Therefore, different redundancy concepts are supported to ensure to avoid failures and an increase availability of NoSQL database systems. (Francis, 2019)

For all non-relational data management approaches the term NoSQL in used to satisfy the criteria listed below:

i. The information is not saved in tables
ii. The SQL language is not the database language.

However, relatively database technology with NoSQL technology needs to be expanded to produce global access that is constant to the services it provides for far-reaching Web applications or applications handling big databases. (Ali, 2019)

Understanding "Big Data" tool is in the concept of data generation is useful, as it helps to store and manage boundless amounts of information generated every second, every day. Although there have been relational database management systems (RDMS) for fixated data storage for some years, scalability, consistency, system efficiency, data collection and data extraction integration have not yet been tackled.
Big Data technologies are tools that enable us to acquire more meaning from data-machine learning, and those that allow us to preserve higher data volumes at greater granularity than ever before. These Big Data technologies were pioneered by Google, and they found their way in the form of Hadoop into the broader IT community.
NoSQL works to help deal with big data volume, variety, and velocity requirements (Andreas & Michael, 2019) explained that Big Data still does not have a binding definition. However many data experts will agree on three V's (3V's): ***V***olume for extensive data volumes, ***V***ariety multiple formats, structured, semi-structured, and unstructured data, and ***V***elocity for high speed and real-time data processing.
(Nzar & Dashne 2019) categorized the features of Big Data into "5Vs": volume, velocity, variety, veracity and value as depicted below:

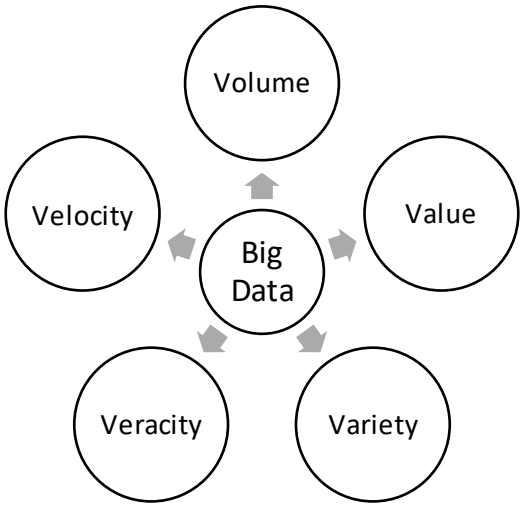

Fig.1: Big Data 5 V's (Source: Nzar & Dashne, 2019)

1. Volume: Shows massive amounts of data, such as data for mobile devices, used for different functions.
2. Velocity: specifies the rate or frequency of generation, updating, processing and access to data.
3. Variety: data accessed through various types of devices, such as videos, photographs, etc.

4. Value: explains how to draw useful knowledge from massive data sets. The most important aspect of any big data tool is value, as it enables the generation of valuable knowledge.
5. Veracity: Refers to a huge precision and value of information.

(Khan, et al, 2017) classifies 1 - 6 v's of big data, which evolves into data worth, making it 7 V's of big data. The seven V's are – Volume, Velocity, Variety, Variability, Veracity, Visualization, and Value.

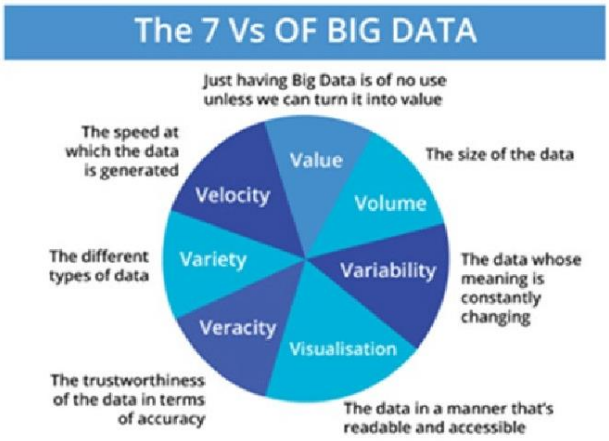

Fig. 2: The seven V's of Big Data (Source: Syed et al, 2019)

II. Review of Related Works

Kalid et al (2017) The paper highlighted literature tackles the issues and finds solutions through the use of contrasting Big Data approaches to the management of No SQL databases; BigTable, DybanoDB and Cassandra. The work also outlines that large companies who do not have the correct storage systems do not need to store and manage big data. The study found that BigTable from Google and DynamoDB from Amazon are critical and profitable for themselves and Casandra is combination of both systems.

Md. Razu et al (2018) studied NoSQL databases for the processing of Big Data, including its transactional and structural issues. The work also highlighted research directions and challenges related to the processing of Big Data which the study believed the information to be incredible to review literature on the NoSQL Big Data Database, including structural data, problems and methods to collect useful information measured in real time.

Vahid (2016) in Big Data: Now and then, opined that growth of data and knowledge is dependent on its availability in the hands of consumers. Which is simple, as cellular phones, laptops, PCs, and more can be easily accessed. In this event, knowledge growth is quickly overflowing—as new content is created by users themselves so that the necessity of managing, using, classifying, and securing such data should be met. The paper attempts to organize and analyze the processes, potential problems, research and initiatives Big Data and display this intensely increasing phenomenon from a prospective horizon.

Ali et al (2019) submitted that NoSQL is altered by most used relational databases for data storage, but it does not completely substitute the SQL, as the name indicates. The paper discusses SQL and NoSQL databases, contrast of conventional SQL with Big Data Analytics NoSQL databases, NoSQL data templates, data storage forms NoSQL, the features and features of each data storage, NoSQL and RDBMS advantages and disadvantages.

III. Structured Query Languages (SQL)

SQL (structured Query Language) on the other hand is traditionally the most famous databases from the early 1970s. An example is Relational databases (accessed by using SQL – e.g. MySQL) where data is stored in a table having rows and columns. Developers back then majorly implemented their designs following the waterfall software development model. This means that every phase of the software development is well planned before the development begins by using a thorough complex entity-relationship ensuring that all that is needed in the database has been carefully thought of and provided (Schaefer, 2015). Even though relational databases were much useful, there were challenges poised to software developers such as if there is to be a slight improvement in the development cycle of the software, developers would have to struggle to keep the project below the stipulated budget and the software might fail users' needs. Another reason is that there has been a spike in the amount of data produced, which is due to a new type of database NoSQL.

A new management approach is considered necessary to support applications such as real-time review of log files, e-commerce transactions and data posted to social media that are enormous in volume. An alternative approach must be put into practice to handle this phenomenal rise in data produced and to get passed all those above-mentioned challenges. The databases introduced, which are the NoSQL databases, notwithstanding do possess some degree of ineffectiveness due to large data production volumes and a lack of support for ACID properties.

IV. Big Data and Databases

Big data can be stored using both Structured databases (MySQL a relational database) and Unstructured databases (MongoDB a non-relational database). Taking into consideration the variation in response time of each database type, different algorithms need to be analyzed to enhance the performance in system monitoring real-time with respect to both SQL and NoSQL updating and inserting big data.

NoSQL database systems are called web-based storage system as long as they fulfill the following requirements:

- Model: The fundamental database model is not relational.
- At least 3 Vs: A large amount of data (volume), flexible data structures (variety) and real-time processing are included in the database system (velocity).
- Schema: A set database schema is not bound by the database management system.
- Architecture: The database architecture supports horizontal scaling and fully distributed web applications.
- Replication: The system of database management supports the replication of data.
- Consistency assurance: consistency can be assured with a delay to give priority to high availability and tolerance of partitions. (2019 Andreas & Michael)

V. Big Data NoSQL Architecture

In comparison to 'Not SQL', NoSQL means 'Not Only SQL' as many consider it to be a type of database that helps to perform big data operations and store them in a valid format. It is commonly used because of its simplicity and wide range of services. Its architectural pattern provides a logical way for data to be processed on the database. (Nzar&Dashne,2019)

NoSQL features:

- Free schema
- Ultimately consistent (as in the BASE property)
- Replication of stores of data to eliminate a single point of failure.
- Capable of processing a range of data and large volumes of data.

VI. Architecture Patterns of NoSQL

The data stored in NoSQL follows any of the four data architecture patterns.

a. Key-Value Store Database
b. Column Store Database
c. Document Database
d. Graph Database

A. Key-Value Store Database

One of the most basic NoSQL database models is this model. The data is collected in the pattern of Key-Value Pairs, as the name implies. A series of strings, integers or characters is typically the key, but it can also be a more advanced form of data. Usually, the value is connected or co-related to the key. The databases for key-value pair storage typically store information as a hash table where each key is unique. The value may be of any form (JavaScript Object Notation (JSON), Binary Large Object (BLOB), strings, etc.). This style of architecture is commonly used in shopping websites or e-commerce applications and its important assets is it's ability for wide management of data volumes, heavy loads and the ease with which keys are used to retrieve data.

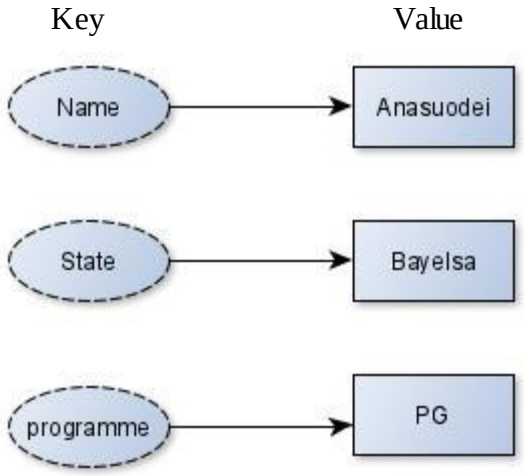

Fig 3. An example of Key-Value

Constraints associated with the key-value store databases is its complexity in handling queries which will attempt to include many key-value pairs that may delay output and may cause data to clash with many-to-many relationships.
Examples here are:

- DynamoDB (developed by Amazon)
- Berkeley DB (developed by Oracle)
- REDIS: An advanced open-source key-value store, also referred to as a data structure server because keys can include strings, hashes, lists, sets and sorted sets. This product, written in C/C++, is searingly quick, which makes it perfect for data collection in real time.
- Riak: An open source that is powerful, distributed database that predictably scales capability and simplifies creation by prototyping, developing, and deploying applications quickly. Written in Erlang and C this technology gives transparent fault-tolerant/fail-over functionality, a comprehensive and versatile API perfect for point-of-sale and factory control systems.

- VoltDB: scalable database in memory that offers complete transactional ACID consistency and ultra-high throughput, self-referred to as the NewSQL. This technology relies on segmentation and replication to achieve high-availability data snapshots and durable command logging using Java stored processes (for crash recovery), making it ideal for capital markets, digital networks, network services, and for online gaming.

B. Column Store Database

This pattern employs data storage in individual cells that is further divided into columns, rather than storing data in relational tuples. Databases that are column-oriented operate only on columns. They together store vast quantities of data in columns. The column format and titles will diverge from one row to another. Each column is handled differently, but still, like conventional databases, each individual column will contain several other columns. (Niharika ,2020)

Basically, columns are in this sort of storage mode. Data is readily available and it is possible to perform queries such as Number, AVERAGE, COUNT on columns easily.

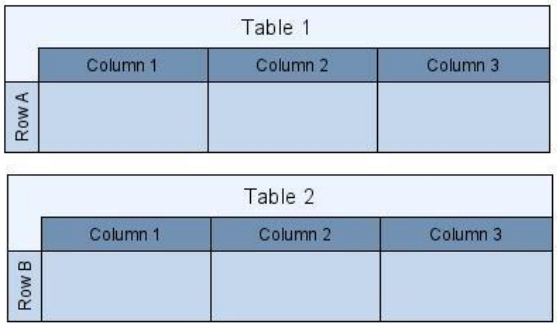

Fig 4. An Example of Column Store

The setbacks for this system includes: transactions should be avoided or not supported, queries can decrease high performance with table joins, record updates and deletes reduce storage efficiency, and it can be difficult to design efficient partitioning/indexing schemes.

Examples here are:

- HBase: HBase is a distributed, portable, Big Data Store modelled after Google's BigTable technology, the Hadoop database.
- Google's BigTable
- Cassandra: An open-source distributed database management system built to manage very large volumes of data scattered over several servers without a single point of failure while delivering a highly accessible service. Written in Java, this product is best for non-transactional real-time data analysis with linear scalability and proven fault-tolerance combined with column indexes.

C. Document Database

In the form of key-value pairs, the record database fetches and accumulates information, but here the values are called documents. A complicated data structure can be represented as a text. The document can be in text form, arrays, strings, JSON (JavaScript Object Notation), XML (Extensible Markup Language) or any other format. The use of nested documents is immensely popular. It is highly efficient since most of the generated information is generally in the form of JSONs and is unstructured.

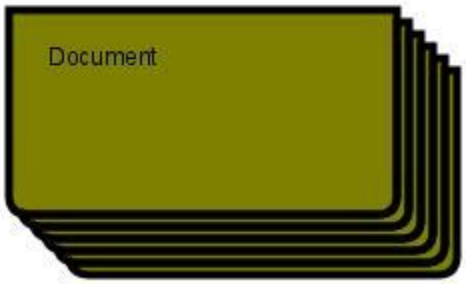

Fig 5. An Example of Document

This format is extremely useful and appropriate for semi-structured data, and it is simple to retrieve and handle documents from storage. The drawbacks associated with this system includes the challenging factor of handling multiple documents and the inaccurate working of aggregation operations.
Examples of such databases are:

- MongoDB: This scalable, high-performance, open-source NoSQL database features document-oriented (JSON-like) storage, full index support, replication, and fast on-site updates from "humongous". This product is suitable for dynamic queries, dynamic data structures, written in C/C++, and if you favour indexes over Map/Reduce.
- CouchDB: Also an open-source database that focuses on the ease of data storage in a series of JSON documents, each with its own definitions of the schema. Eventual consistency is enforced by ACID semantics that prevents locking database files during writing. This product, written in Java, is suitable for web-based applications that manage large quantities of data that are loosely organized.

D. Graph Database

This pattern of architecture clearly deals with information storage and management in graphs. Graphs are essentially structures that represent relations between two or more objects in some data. Objects or entities are referred to as nodes and are connected with relationships known as edges. There is a unique identifier on each edge. For the graph, each node serves as a point of touch. In social networks where there are many and large numbers of entities, this pattern is very widely used and each entity has one or many characteristics that are linked by edges.

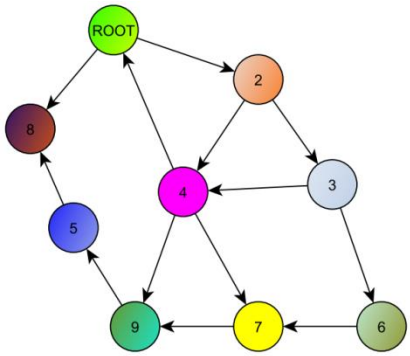

Fig 6. An Example of Graph

There are loosely connected tables in the relational database pattern, whereas graphs are often strong and rigid in nature, have a faster traversal due to connections, and allow spatial data to be easily handled, but incorrect connections can lead to infinite loops. (Ian, 2016)

Examples of such databases are:

- Neo4J: The leading native graph database and graph platform is Neo4J: Neo4j. For enterprise levels of security and high performance and reliability by clustering, it is available both as open source and through a commercial license. Cypher, the graph query language of Neo4j, is very simple to learn and can use newly released open source toolkits, "Cypher on Apache Spark (CApS) and Cypher for Gremlin to operate across Neo4j, Apache Spark and Gremlin-based products."
- FlockDB (Used by Twitter): FlockDB is easier than other graph databases since it attempts to solve less problems. It fits horizontally and is optimized for on-line, low-latency, high throughput environments such as websites.
- ArangoDB: this type of graph database requires one database, One Query language, Three models for data. The Limitless Possibilities. ArangoDB is a fast-growing native multi-model NoSQL database, with more than one million downloads.
- OrientDB: OrientDB is the first Distributed DBMS multi-model with a True Graph Driver. Multi-Model means NoSQL 2nd generation that is capable of managing complex domains with amazing efficiency.
- Titan: Titan is a scalable graph database designed for storing and querying graphs spread over a multi-machine cluster comprising hundreds of billions of vertices and edges. Titan is a transactional database that can facilitate the real-time execution of complex graph traversals by thousands of concurrent users.
- DataStax: In a rapidly changing environment where aspirations are strong, DataStax helps businesses thrive and new technologies occur daily.
- Amazon Neptune: Amazon Neptune with a highly connected datasets to create and run applications is secure, fast and has a fully managed graph database service that is easy. (Niharika, 2020).

VII. NOSQL Strength and Weakness

S/N	Strength	Weakness
i.	Suitable, has strength to store and lookup for Big Data.	Requires a conceivably expensive infrastructure
ii.	Is Application focused	Is overly complex
iii.	Supports HUGE data capacity	Engineering talent still hard to find
iv.	Fast Data Ingestion (loads)	Generally, there is no SQL interface
v.	Fast Lookup Speeds (across clusters)	Limited programmatic interfaces
vi.	Enables data streaming and high performance data server not needed	Inadequate for Analytic Queries (aggregations, metrics, BI)

VIII. Conclusion

This work reviewed and studied big data in recent times has and ways to handle increasing exceptionally in volume data. Structured database experiences difficulties when handling unstructured data due to its size. To make sense of Big Data, new architecture and method are

needed, also this work examined the various Big Data NoSQL Databases Architecture, types associated with them, importance, and usage.

References

Ali, W., Shafique, M. U., Majeed, M. A., & Raza, A. (2019). Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. *Asian Journal of Research in Computer Science*, *4*(2), 1–10. https://doi.org/10.9734/ajrcos/2019/v4i230108

Andreas Meier, & Michael Kaufmann. (2019). SQL & NoSQL Databases Models, Languages, Consistency Options and Architectures for Big Data Management. Springer Vieweg. https://doi.org/10.1007/978-3-658-24549-8

Best Graph Databases in 2020. (2020, May 27). G2. https://www.g2.com/categories/graph-databases

Big Data and NoSQL Technologies | DB Best Chronicles. (2012, June 15). DB BEST. https://www.dbbest.com/blog/big-data-nosql-technologies/

Column-Oriented Database Technologies | DB Best Chronicles. (2012, July 24). DB BEST. https://www.dbbest.com/blog/column-oriented-database-technologies/

Francis, K. K. (2019). NoSQL Databases for Big Data Management: Review and Application in Mobile Commerce. *ResearchGate*, 9. https://doi.org/10.13140/RG.2.2.10239.87204

Garrett, Alley. (2019, March 4). *What Is Big Data Architecture? - DZone Big Data*. Dzone.Com. https://dzone.com/articles/what-is-big-data-architecture

Guy Harrison. (2015). Next Generation Databases NoSQL, NewSQL, and Big Data. APRESS.

Ian. (2016, June 16). *What is a Graph Database? | Database.Guide*. https://database.guide/what-is-a-graph-database/

Kalid, S., Syed, A., Mohammad, A., & Halgamuge, M. N. (2017). Big-data NoSQL databases: A comparison and analysis of "Big-Table", "DynamoDB", and "Cassandra". *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*, 89–93. https://doi.org/10.1109/ICBDA.2017.8078782

Khan, M. A., Uddin, M. F., & Gupta, N. (2014). Seven V's of Big Data understanding Big Data to extract value. *Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education*, 1–5. https://doi.org/10.1109/ASEEZone1.2014.6820689

Le, J. (2019, November 20). *An Introduction to Big Data: NoSQL*. Medium. https://medium.com/cracking-the-data-science-interview/an-introduction-to-big-data-nosql-96b882f35e50

Liyakathunisa Syed, Saima Jabeen, S. Manimala, & Hoda Ahmed Galal Elsayed. (2019, January). *(5) Data Science Algorithms and Techniques for Smart Healthcare Using IoT and Big Data Analytics: Towards Smarter Algorithms*. ResearchGate. https://www.researchgate.net/publication/330723399_Data_Science_Algorithms_and_Techniques_for_Smart_Healthcare_Using_IoT_and_Big_Data_Analytics_Towards_Smarter_Algorithms#fullTextFileContent

Md. Razu, A., Mst. Arifa, K., Md. Asraf, A., & Kenneth, S. (2018). A literature review on NoSQL database for big data processing. *International Journal of Engineering & Technology*, *7*(2), 902–906. https://doi.org/10.14419/ijet.v7i2.12113

Niharika, P. (2020, January 2). NoSQL Data Architecture Patterns. *GeeksforGeeks*. https://www.geeksforgeeks.org/nosql-data-architecture-patterns/

Raouf, D & Ali, N. (2019). Improving the performance of big data databases. Kurdistan Journal of Applied Research. 4. 206-220. 10.24017/science.2019.2.20.

Schaefer, L. (2015). What is NoSQL? Available at https://www.mongodb.com/nosql-explained

The basics of NoSQL databases—And why we need them. (2019, January 31). FreeCodeCamp.Org. https://www.freecodecamp.org/news/nosql-databases-5f6639ed9574/

Vahid Rahmati. (2016). Big Data: Now and Then. *International Journal of Emerging Computing Methods in Engineering*, *1*(2), 1–6. https://www.researchgate.net/publication/309458088_Big_Data_Now_and_Then