A Survey of Nosql Database For Analyzing Large Volume Of Data In Big Data Platform

R.S.Raghav	extsuperscript{1*}, J. Amudhavel	extsuperscript{2}, P.Dhavachelvan	extsuperscript{3}

	extsuperscript{1}Department of CSE, KL University, Andhra Pradesh, India.
	extsuperscript{2}Department of CSE, KL University, Andhra Pradesh, India.
	extsuperscript{3}Department of Computer Science, Pondicherry University, Puducherry, India

*Corresponding author E-mail: vpmrags@gmail.com

Abstract

The massive improvement of data in the present era is drastically increased, where millions of data is emerging from variety of applications. Due to this massive flow of data the importance of data becomes a key factor in all applications. In old technology the space occupied by data is very less, but in the present scenario the value of every piece of data plays a vital role. The organization needs some new technology to process large amount of data in an effective way. For that purpose data exploration and visualization systems play a vital role in the Big Data era. It is a complex task for the companies to explore and visualize very large datasets. Every company should follow some protocol to have accurate insight from analysis of large volume of data. This strategy helps organizations to enhance their business functionalities and it also helps to identify the way to improve the quality of their products. The big data [25-29] contains some unique databases to handle massive volume of data. In this survey, we explain the characteristics of database and challenges. It also describes about the different techniques and tools currently used for handling large sets of data and their capabilities to support massive volume of data from variety of data sources.

Keywords: NoSQL Database, Parallel DBMS, Big Data, MongoDB, Column Stores Database, Document Store Databases

1. Introduction

The role of database for collecting the data from the raw data is a difficult process. To handle these problems the organization went to a technique for storing and processing the data which is known as Parallel DBMS. It is defined as a DBMS which works on multiple locations across multiple processor and disks. The parallel way of processing is done to enhance performance of system. The main goal of having parallel database is to increase the processing speed and to produce high scalability for large number of users.

The Parallel DBMS works to overtake the problems faced by single process of systems, where they fail to satisfy some criteria like they are not cost-effective scalability, reliability, and performance [1]. These criteria's are satisfied by a powerful and financially attractive parallel DBMS driven by multiple processors. The Parallel DBMS consists of more links in smaller machines to produce the same throughput as a single, larger machine. This strategy increases scalability and reliability of the system when compared with single-processor DBMSs. The shared resource management is a key source of using the multiple processors in parallel DBMS. The cluster of nodes is also a kind of architecture executed by parallel database system which is known as "shared-nothing nodes"[2]. This strategy is used for improving the high-speed interconnect among the nodes for quick processing. The horizontal partitioning of relational table and the execution of the partition of SQL queries is used by the parallel DBMS.

The concept of horizontal partitioning is the rows distribution in a relational table across the nodes of the cluster is done for executing the parallel processing of data[3]. The multiple CPU is used for the transaction performed by multiple nodes with the help of parallel data manipulation language. The components of the system like memory, disk and processor needs to handle load balancing, by using the parallel database approach the constraints can be easily managed. Even though the parallel DBMS consist of many positive aspects, they fail to tackle large data sets [1,3]. These large datasets are gathered in unstructured format such as audio, video, images etc. In order to handle these types of data, the organization should move to some other effective technology like NOSQL database.

2. NOSQL

In big data environment the term NoSQL places a major role for handling different variety of data. As we know the four main V's of big data like (Volume, Variety, Velocity and Value) are the backbone for handling large data sets. The characteristics of NoSQL are it has the ability to handle large data sets which are widely spread across the globe with different database technologies [4]. The main goal of using NoSQL database is, the response and performance of the data in Nosql database is very effective when compared to relational databases.
3. Types NoSQL Database

The NoSQL database is classified into different types of database and this is done by the 4S of data collected from different sources. Some of the Nosql database is: 1) Column Store Database 2) Document Store Database 3) Graph Store Database 4) Key-Value Store Database 5) Object Store Database 6) Multimodal Store Database 7) Cloud Store Database 8) XML Store Database 9) Multidimensional Store Database 10) Event Streaming Store Database [5,6].

1. Column Store Database

From the name we can understand the concept of using column based DB. It stores huge volume of datasets in columns instead of using rows. The comparison of each column with rows in an RDBMS table and the identification of row are done with the help of key assigning, where the rows can have multiple columns in it [7]. The adding of column can be carried to any row at any time and the column mapping is done for easy storing. Some of the famous column database is Cassandra, Hyper table, Accumulo and HBase etc.

2. Document Store Database

It consists of document which comprises of pair each key with a difficult structure [8]. The document database is used for storing and retrieving the documents, which is stored in different format such as XML, JSON etc. The hierarchical tree data structure is used for storing these documents. It also consist of maps, scalar values and collections, and the process of storing of documents can be similar to each other document [9]. Some of the document databases are Mongo DB, Couch DB, JSON ODM and TerraStore etc.

3. Graph store Database

The idea of using the graph store database is to handle the data in an accurate way by carrying the information using network [9]. The nodes are defined as the instance of an object and relations are referred by the edges. Each node has its own relationships and it also allows represents both relationships between the domain entities. The secondary relationships are referred as path, tress and category for carrying partial indexing. There is no limitation for relationships of a node and they can be represented in the same graph database. The graph store Database is Neo4J, Graph Base, Hyper Graph DB and Infinity Graph etc [10].

4. Key-Value Stores

The simplest and easy way of storing data in NoSQL databases is key value DB. The key or attribute name for storing each item in the database with the value of the each item [12]. The user can get the information like by knowing the key value and it is mainly used for just storing purpose. The primary key is used for easy accessing of data and also for storing purpose. Some of the familiar key value store database is voldermort, Amazon DynamoDB, Berkeley DB, Riak and Azure Table storage [11, 13].

5. Object Store Database

It also known as objects database management systems; here Objects are used as attributes and methods for data storage. The attributes describes about the characteristics of an object like such as integers, strings, and real numbers or complex object [14]. The functionality of an object is referred to the methods. Some of the object store database are Velocity DB, HSS DB, Magma, EyeDB and object DB etc.

6. Multimodal Store Database

The data model consists of some process like collecting the data from raw data, organizing the data, storing and processing the data [15]. This type of database supports multiple data models against a single and it is integrated according to the requirements. Some of the multimodal store DB is Arango DB, orient DB, wonder DB, Cortex DB and gunDB.

7. Cloud Store Database

The cloud store databases are mostly used in the cloud environment. It provides service to the user by storing the data in their respective database [16]. It has the ability to achieve optimized scaling, providing high availability, multi-tenancy with effective resource allocation. Some of the cloud store database is Oracle Coherence, Hazelcast and Crate data etc.

8. XML Store Database

These database are used for storing XML based data format and the properties of using XML DB is for fast query processing of the data in the xml format [17]. The XML store DB is EMC DocumentsDB, Sedna, BaseX etc.

9. Multidimensional Store Database

It is also known as multidimensional database management system (MDDBMS), it has the ability to execute the data processing in an effective and easy way[18]. Some of the MDDBMS is Dagger DB, Intersystems Cache,Globals and Minim DB.

10. Time Streaming Store Database

The data are dynamic in nature and nature of data is changed within milliseconds. Inorder to process these types of data, the organization need Time streaming DB. It is used for storing and analyzing high-frequency time-series data at scale. The Time streaming DB is AXIbase and Riak TS etc[19].

4. Merits of NOSQL

The NoSQL have more benefits, when compared to relational database. They are highly scalable and it can show better performance with low cost. Some of the benefits of using Nosql database are displayed below

1. Dynamic Design

The NOSQL db does not follow any design. In rdbms selection of schema is done before inserting the values in it[20]. But in Nosql database there is no need of any schema process. The user can have quick process and retrieving of the data is carried in short span of time.

2. Scalability

They are highly scalable in nature because they don't have any procedure like scale up and scale out for handling more number of loads. The scale up process can be fit for large volume of data, but the Nosql can be built with an ability of extensible at the time of increase in the load [17].

3. Easy Placement of server
The NOSQL is completely suitable for executing unstructured data in an effective way. And it become expensive and the performance will become slow. The horizontal placement of server is used for handling massive volume of data rather than using the server vertically [19]. This strategy used for prohibiting server failure and the replication process is carried in an effective way to stop data failure.

4. Cost Effective

The main benefit of using Nosql is cost effective, where it doesn't need any special requirement of hardware for carrying large data sets[21]. It has the capacity for carrying the large data sets. The client doesn't require any complex process because of it simple nature.

5. Flexibility

The user can create flexible data models in the Nosql database and they don't have any restrictions for creating the schema [22]. The large data sets can be handled in an effective way without any issues.

6. Replication

The process of replication plays an important role in every database, because of processing more volume of data. Inorder to avoid failure the DB should have a backup plan to save and retrieve data from crash or dump in the database [23].Most NoSQL databases also support automatic replication, and there is no need of any special server or hardware for carrying replication process. The above mentioned are some of the important merits of using Nosql database for handling more volume of unstructured data.

5. Demerits of NOSQL

The nature of the Nosql database has more benefits, where the user can use this database for every aspect. But they also face some obstacles, which should be avoided before incorporating the database in the system.

1. Skilled Person

The query processing RDBMS is old and the person in organization have deep understand and knowledge about the rdbms. Even though Nosql is easy for implementation it should require some skilled person to know or to have deep insights about the nature of data, inorder to get the effective output [19].

2. Support

The Nosql DB is an upcoming technology for handling big data sets so the support is less when compared to RDBMS. If there is any presence of problem in RDBMS the vendors can easily get the support from the organization.

3. Displaying of Result

It is a major issue in Nosql because once the data is analyzed the displaying of results should be done in an effective way [24]. The data visualization is a different are in big data environment for displaying the results the company should use tools to analyse the results. This process is difficult, if organization fails to choose a proper data visualization tool. The growth of unstructured data is gradually increased, so the company or organization needs to move the database which can solve variety of problems in an effective way. That is the reason most of the organization is moving for Nosql database in big data environment. The fig 1 describes about the classification of NOSQL database. The Table 1 refers to the comparison of variety of NOSQL database used in the big data environment.

6. Discussion

In this section we discuss which type of big data DB is used by the companies to analyze large volume of data. In fig 2 the availability of the column store No Sql database is shown, where bar chart explains the availability of Database to handle massive volume of data type when compared to other type. The usage of multidimensional store and object store database grab the next high level of availability. To handle massive volume of data, the database should have a nature of fault tolerant. If there is any presence of any fault or error, the database should have a backup strategy to tackle the problem. The fig 3 give an idea about the role of key value store and column store NoSQL db for handling fault tolerant problem and this two NoSQL DB was used more when compared with other DB. The performance is one of the key factors, to know about the strategy of every NoSQL DB to handle high amount of variety of data. In fig 4 the performance of each NoSQL DB was shown and the key value DB has the high performance which is also known for its simplest and easy way of storing data in NoSQL databases other type of NOSQL DB such as object store and column store DB falls in second and third position, this is because most of the data can be analyzed easily and efficient way by using the key vale store DB than others. The scalability plays a vital role in every DB; the role of NOSQL DB is mainly used for handling more number of data and high number of users. In Fig 5 the Scalability of Nosql Db Was Shown and The object store DB is used to have high scalability when compared to other NOSQL DB. Even though other NoSQL DB falls close to the object DB, the usage of more number of users will be efficiently carried by using the object store value DB.
Fig. 1: Classification of NoSQL Database
Table 1: Comparison of NOSQL Database

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>Dagger DB</th>
<th>Intersystem Cache</th>
<th>Globals</th>
<th>Minim DB</th>
<th>AXIbase</th>
<th>Riak TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Avg</td>
<td>High</td>
<td>Avg</td>
</tr>
<tr>
<td>Written In</td>
<td>C#</td>
<td>Java / .NET</td>
<td>Java</td>
<td>R, Java, Ruby, Python</td>
<td>Java</td>
<td>C#</td>
</tr>
<tr>
<td>Query Method</td>
<td>LINQ queries</td>
<td>JDBC, ODBC</td>
<td>Squeak</td>
<td>MUMPS-based query</td>
<td>SQL, Gremlin, SparQL</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Avg</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Scalability</td>
<td>High</td>
<td>Avg</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Fault Tolerant</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
</tr>
</tbody>
</table>

Table 2: Comparison of NOSQL Database

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>HBase</th>
<th>Membase</th>
<th>DynamoDB</th>
<th>CouchDB</th>
<th>Berk DB</th>
<th>Hypergraph</th>
<th>Iorra</th>
<th>Amazon DynamoDB</th>
<th>Berk DB</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Avg</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Avg</td>
</tr>
<tr>
<td>Written In</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>C#</td>
</tr>
<tr>
<td>Query Method</td>
<td>CQL</td>
<td>SQL, Gremlin</td>
<td>SQL</td>
<td>SQL</td>
<td>SQL</td>
<td>SQL, Gremlin, SparQL</td>
<td>SQL</td>
<td>No query method</td>
<td>LINQ</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Avg</td>
</tr>
<tr>
<td>Scalability</td>
<td>High</td>
<td>Avg</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Fault Tolerant</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Comparison of NOSQL Database

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>HBase</th>
<th>Membase</th>
<th>DynamoDB</th>
<th>CouchDB</th>
<th>Berk DB</th>
<th>Hypergraph</th>
<th>Iorra</th>
<th>Amazon DynamoDB</th>
<th>Berk DB</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Avg</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Avg</td>
</tr>
<tr>
<td>Written In</td>
<td>C#</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>Java</td>
<td>C#</td>
</tr>
<tr>
<td>Query Method</td>
<td>CQL</td>
<td>SQL, Gremlin</td>
<td>SQL</td>
<td>SQL</td>
<td>SQL</td>
<td>SQL, Gremlin, SparQL</td>
<td>SQL</td>
<td>No query method</td>
<td>LINQ</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Avg</td>
</tr>
<tr>
<td>Scalability</td>
<td>High</td>
<td>Avg</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Fault Tolerant</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td>Avg</td>
<td></td>
</tr>
</tbody>
</table>
7. Conclusion

The value of data in present world is high, where a normal database couldn’t have the capacity to handle massive volume of data. The handling of data and analyzing the large volume of data need to carry pre process like proper extraction of data from variety of data sources. They should know the 3v’s of data such as volume, variety, velocity and value etc. According to it, the company should select the proper database, process, scripting language and last the proper data visualization tool for carrying effective analyses. In this paper we discussed about the variety of database and its features. These strategies help the business people to know the value of each data and how to process the data and analyze it and how to improve their business value. By selecting a perfect database the company can gain deep insights about the data and quick decisions on the advantageous business opportunities can be done effectively.

References