hid323

by Hid323 Hid323

Submission date: 17-Nov-2017 01:01PM (UTC-0500)
Submission ID: 881714204
File name: hid323.pdf (249.55K)
Word count: 1886
Character count: 9189

NoSQL Databases in Support of Big Data and Analytics

Uma M Kugan
Indiana University
711 N Park Avue
Bloomington, IN 47408, USA
umakugan@iu.edu

ABSTRACT

The data volume is increasing at high velocity and it comes from various sources with different formats. These data no longer fits into defined structure and hence the need for handling the big data using NoSQL. This paper will highlight on what is NoSQL and where and when it should be used for and also why Big Data can not be handled in traditional RDBMS.

KEYWORDS

i523, hid323, NoSQL, Bigdata, RDBMS

1 INTRODUCTION

RDBMS have always been the preferred method of storage for many years and its powerful Query language made it very user friendly. Data has grown exponentially in a past decade due to the growth of social media, e-commerce and web applications which posed a big challenge for the traditional databases. Need of the hour is not just to limit the data within the structure, but also ability and flexibility to read and store data from all sources and types, with or without structure. Companies that has larger amount of unstructured data are shifting away from traditional relational databases to NoSQL [?]. There are lot of limiting factors in these databases for Big Data especially Structured schema which was one of the main reason for RDBMS to scale it for larger databases [?].

2 LIMITATION OF RDBMS

Choice of database chosen depends on their data model, data access and data latency. But in this era, every organization needs all three at the same time and which can not be provided by traditional databases.

Scalability - RDBMS are designed for scaling up meaning if storage needs to be increased, we need to upgrade other resources in the existing machine whereas in NoSQL we just have add additional nodes in the existing cluster.

Acid Compliance - RDBMS are always acid compliant and which of course is its strength to process transactional data while the drawback is it can not handle larger volume of data without impacting the performance. If there are use cases where we do not require ACID compliance and where it has to handle huge volume of data in significantly very less time, then NoSQL is the solution.

Complexity - RDBMS stores the data in defined, structured schema in tables and columns. If the data can not be converted to store in tables, it becomes cumbersome to handle such situations.

Time Consuming - Analyzing data in real time is highly impossible with RDBMS and no one have time to wait for longer load schedules in traditional way of data warehouse and ETL.

3 NOSQL

"The term NoSQL was first used by Carlo Strozzi to name a database management system (DBMS) he developed. This system explicitly avoided SQL as querying language, whi it was still based on a relational model" [?]. The term NoSQL means that the database does not follow the relational model espoused by E.F Codd in his 1970 paper, "A Relational Model of Data for Large Shared Data Banks which would become the basis for all modern RDBMS" [?]. NoSQL does not mean NO to SQL. It means Not Only SQL. NoSQL means storage is just nonvolatile object store with no maintenance concerns. Most NoSQL Databases are open source which allows everyone to evaluate the tool of their choice at low cost. NoSQL databases, because of it's simpler data model, it does not need DBA's to maintain the health of the database. NoSQL databases are widely used in big data and in real-time applications.

4 NOSQL TYPES

In Edlich et al. identify four classes of NoSQL systems as 'Core-NoSQL' systems: Key-Value stores, Wide column stores, Graph databases and Document stores [?].

Key-Value Stores - Key is the unique identifier or label of an item whose data or its location is stored in the value. It is very basic non relational data types which is most commonly used. Example include Redis, Amazon DynamoDB and Oracle NoSQL.

Wide Column Stores - Every record in the stores may differ in the number of columns. This is very important factor for analytic because it needs very low I/O and also reduces the volume of data that are read to the disk. It is also known as tabular NoSQL database.Examples include HBase, Google BigTable and Cassandra.

Graph Database - As the name indicates, it uses graph structures nodes and edges to represent the data. This is very useful in depicting social relationship, network topology. Examples include Neo4J and DataStax Enterprise Graph.

Document Stores - It stores the data as document typically in JASON or XML format. It is very flexibile and one can easily access the data. This is used widely in many places. Examples include MongoDB and CouchDB.

5 ADVANTAGES OF NOSQL

NoSQL databases differ from traditional databases in features and functionality. There is no common query language, high I/O performance, horizontal scalability and do not enforce schema. RDBMS scales up vertically making single CPU works faster and performance can be increased adding extra CPU or RAM, whereas a NoSQL database scales horizontally by making many CPUs works

together and also by dividing the jobs into multiple chunks [?]. It is very flexible and let the users to decide to use the data the way they want. Data localization in NoSQL databases is acheived by distributing it across many geographic regions. NoSQL databases does not have need specific applications or hardwares to implement replication [?]. Since NoSQL does not enforce atomicity and hence it is not reliable where data accuracy is very critical. The main advantage of NoSQL is its data is always replicated on each node and so the data is always available and there is zero downtime. RDBMS supports master-slave architecture so data can always be written to master and read only data is available in all slave machines whereas in NoSQL, both read and writes are enabled in all nodes. In general most of the NoSQL databases performance is better than SQL databases.

6 NOSQL CHALLENGES

NoSQL databases have created lot of interests in each organization to move away slowly from traditional databases but there are many challenges to overcome. RDBMS are much more matured, been around for many years and the best technical support is available. So there is always fear of unknown until the technology gets widely accepted and used. Most of the NoSQL databases are open source and support and reassurance that any organization gets from their traditional RDBMS vendors are challenged. Even though NoSQL goal is to provide no admin solution,in current trend, it requires lots of skills to maintain and learn. It is highly tempting for any organization to adopt living edge technology, but that adoption needs to be embraced with selection of best tool and with extreme caution [?]. Ad-hoc query analysis is quite complex in NoSQL databases and it requires expertise to write even a simple query.

7 NOSQL FOR BIG DATA

When to choose NoSQL over an RDBMS depends on ACID (Atomicity, Consistency, Isolation, Durability) vs BASE (Basically Available, Soft state, Eventual consistency) and also on the type of the data that the organization is dealing with. Based on the project requirements, If the real time updates is needed to perform data analytics, NoSQL is the solution for applications that receives large volume of data in a real time and where data insights are generated using real time data that was fed. NoSQL is the best fit where the enterprise does not require complex messaging features for publishing/subscribing. NoSQL comes handy where data structure is not restricted by schema(schema less design). Many NoSQL database compromises consistency over availability and data partition.

8 HOW TO HANDLE RELATIONAL DATA IN NOSQL

NoSQL database in general can not perform joins between data structures and hence the schema has to be designed in such a way so that it can support joins [?]. Below are the key things that needs to be considered to handle relational data in a NoSQL.

Avoid Sub Queries : Instead of using complex sub queries or nested joins to retrieve the data, break into multiple queries. NoSQL performances are very high when compared to traditional RDBMS Queries.

Denormalize the Data : For faster retrieval of data, it is essential to compromise on denormalizing the data rather than storing only foreign keys.

9 RDBMS TO NOSQL MIGRATION

Database Migrations are always cumbersome and it is better to plan well ahead and take an iterative approach. Based on the need of application, one have to choose which NoSQL database we are going to migrate to [?].

9.1 Planning

The goal of any migration should be better performance at the reduced cost with the newest technology. While migrating from RDBMS, we have to consider volume and source of data that is going to be migrated to NoSQL. All the details should be documented well so that we do not have to face unplanned surprises at the end.

9.2 Data Analysis

This is very critical and will help in understanding the nature of the data and how that data is accessed within the application. Based on the analysis of data usage, we will be able to define how data will be read/written which will help us in building a better data model.

9.3 Data Modeling

When migrating from any RDBMS, depending on the need of application, we may have to sometimes denormalize the data. In this phase, based on the data analysis and the tech-stream, we have to define keys and values.

9.4 Testing

Testing is always very critical and crucial for any migration projects. We have to define all possible test cases and different types of testing: unit, functional, load, integration, user acceptance and smoke testing have to be performed and outputs have to be clearly documented.

9.5 Data Migration

Once all the above steps are successfully tested and implemented, next final act is to migrate all data from RDBMS to NoSQL. Post implementation validation has to be carried out to make sure everything went well as per the plan and it has to be monitored for few days until the process is stabilized. If there are any issues with the migration, rollback to original state and root cause analysis have to be performed to identify and fix the issue. Once issue has been fixed, data migration has to be scheduled and this step goes in cyclic unless migration was completely successful.

10 CONCLUSION

With the explosion of the data in the recent years, have paved the big way for the growth of Big Data and everyone wants to move their applications and data into Big Data. Building a big data environment is relatively very cheap when compared to migrating the existing data in RDBMS to NoSQL. We have to carefully weigh in, understand the data and how the data will be used in the use case to enjoy the full benefit of migrating into No SQL.

ACKNOWLEDGMENTS

My sincere thanks to my mentor and leader Vishal Baijal and to my colleague Michael Macal for their support and suggestions to write this paper and also to my fellow classmate Andres Castro Benavides for his support. My special thanks to Dr. Gregor von Laszewski for his support in fine tuning the paper.

hid323

ORIGINALITY REPORT

5%	5%	1%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	sewiki.iai.uni-bonn.de Internet Source	3%
2	d0.awsstatic.com Internet Source	2%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		