

Comparative Study of SQL and NoSQL Databases to evaluate their suitability for Big Data Application

¹Ms. Deepika V. Shetty, ²Ms. Sana J.Chidimar

^{1,2}ASM Institute Of Management & Computer Studies (IMCOST) C-4, Wagle Industrial Estate, Near Mulund Check Naka, Thane (W), Mumbai – 400604, Department of MCA, University Of Mumbai, India

Abstract: Big data application has become an imperative for companies across a wide variety of industries. One of the critical decisions facing companies embarking on big data projects is which database to use, and often that decision swings between SQL and NoSQL. SQL programming language have long been the top -- and, in many cases, only --choice of database technologies for organizations. SQL has already earned its stripes in large organizations and big data is just one more job that powerfully built. NoSQL is SQL has the impressive track record, the large installed base, but NoSQL is making impressive gains and has many proponents. NoSQL is increasingly being considered a possible alternative to relational databases, especially for Big Data applications. SQL & NOSQL databases and tries to answer which of these is better for big data application in terms of its performance, scalability, flexibility and many more.

Keywords: Big data, SQL, NO SQL.

I. INTRODUCTION

In today's world rapid growth of computer and internet causes an efficient storage and retrieval of data. Big data requires exceptional technologies to efficiently process large quantities of data within tolerable elapsed times. As big data is explode in many companies. One of critical decision facing companies on bigdata project is which database to use Sql or Nosql. So currently most industry experts prefer to work with both as the need requires. SQL is structured query language . SQL enables increased interaction with data. It also allows a broad set of questions to be asked against a single database design. That's key since data that's not interactive is essentially useless, and increased interactions lead to new insight, new questions and more meaningful future interactions. SQL is standardized, allowing users to apply their knowledge across systems and providing support for third-party add-ons and tools. SQL is orthogonal to data representation and storage. Some SQL systems support JSON and other structured object formats with better performance and more features than NoSQL implementations.

When we talk about Big Data in the NoSQL space, we're referring to reads and writes from operational databases – that is, the online transaction processing that people interact with and engage in on a daily basis .Operational databases are not to be confused with analytical databases, which generally look at a large amount of data and collect insights from that data.

While the Big Data of operational databases might not appear to be as analytical when scratching the surface, operational databases generally host large datasets with ultra-large numbers of users that are constantly accessing the data to execute on transactions in real time. The scale to which databases must operate to manage Big Data explains the critical nature of NoSQL, and thus why NoSQL is key for Big Data applications. Data is becoming increasingly easier to capture and access through third parties, including social media sites. Personal user information, geographic location data, user-

generated content, machine-logging data and sensor-generated data are just a few examples of the ever-expanding array being captured. Enterprises are also relying on Big Data to drive their mission-critical applications. Across the board, organizations are turning to NoSQL databases because they are uniquely suited for these new classes of data emerging today.

II. SQL DATABASE

SQL databases are primarily called as Relational Databases (RDBMS). SQL is equally effective at running blazingly fast ACID transactions. The abstraction that SQL provides from the storage and indexing of data allows uniform use across problems and data set sizes, allowing SQL to run efficiently across clustered replicated data stores. Structured Query Language (SQL) is a proven winner that has dominated for several decades and is currently being aggressively invested in by big data companies and organizations such as Google, Facebook, Cloudera and Apache.

Relational databases having the variety of limitations due to constant growth of stored and analysed data, e.g. the restrictions on scalability and storage, and efficiently losing of query as the volume of data is very large, and the storing and managing of larger databases become challenging.

Google, Amazon, Facebook, and LinkedIn are among the first companies to discover the serious limitations of SQL database technology for supporting big data and big user's requirements.

III. NoSQL DATABASE

NoSQL Database, also known as "Not Only SQL" is an alternative to SQL database which does not require any kind of fixed table schemas unlike the SQL. NoSQL generally scales horizontally and avoids major join operations on the data. NoSQL database can be referred to as structured storage which consists of relational database as the subset. NoSQL Database covers a swarm of multitude databases, each having a different kind of data storage model. The most popular types are Graph, Key-Value pairs, Columnar and Document. NoSQL is a database technology driven by Cloud Computing, the Web, Big Data and the Big Users. NoSQL now leads the way for the popular internet companies such as LinkedIn, Google, Amazon, and Facebook - to overcome the drawbacks of the 40 year old RDBMS.

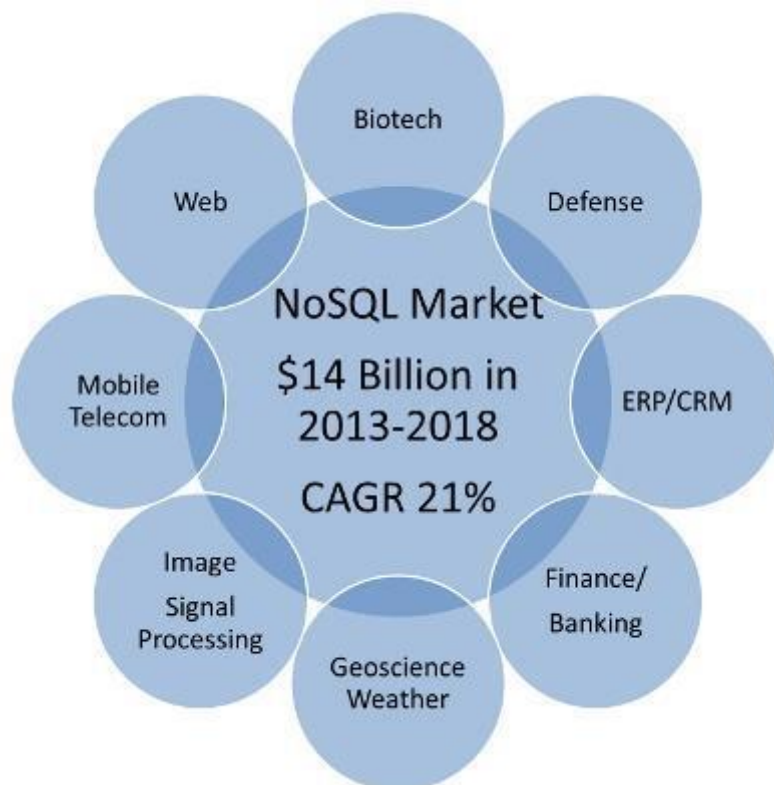


Fig: NoSQL in Big Data Application

HBase for Hadoop, a popular NoSQL database is used extensively by Facebook for its messaging infrastructure. HBase is used by Twitter for generating data, storing, logging, and monitoring data around people search. HBase is used by the discovery engine Stumble upon for data analytics and storage. MongoDB is another NoSQL Database used by CERN, a European Nuclear Research Organization for collecting data from the huge particle collider “Hadron Collider”. LinkedIn, Orbitz, and Concur use the Couchbase NoSQL Database for various data processing and monitoring tasks.

Overall, with the rise in Web and mobile applications, alongside emerging trends, shifting online consumer behavior and new data classes, the projects the industry is working on require a database technology that is capable of providing the scalable, flexible solution to manage and access data. NoSQL technologies are the only solution available to effectively meet these needs.

Couchbase is a NoSQL database technology provider and the company behind the couchbase project. Couchbase Server, the company’s flagship product, is a NoSQL document-oriented database with production deployments at Amadeus, AOL, Cisco, LinkedIn, Orbitz, Salesforce.com, Viber and hundreds of other enterprises worldwide. Couchbase is known for its easy and reliable scalability, consistent high performance, 24x365 availability, and flexible data model for ease of development. Couchbase is headquartered in Silicon Valley, and is funded by Accel Partners, Ignition Partners, Mayfield Fund and North Bridge Venture Partners.

IV. COMPARING SQL OR NoSQL BETTER FOR BIG DATA APPLICATION

1. Enables Interaction:

As SQL is a declarative query language it enables interaction. By contrast, NoSQL programming innovation MapReduce is a procedural query technique. MapReduce requires the user to not just know what they want, but additionally requires them to state how to produce the answer. There is technical difference with two critical reasons.

First, declarative SQL queries are much easier to build which opens up database querying to analysts, operators, managers and others. Second, abstracting what from how allows the database engine to use internal information to select the most efficient algorithm. Change the physical layout or indexing of the database and an optimal algorithm will still be computed. In a procedural system, a programmer needs to revisit and reprogram the original how. This is expensive and error-prone.

2. Speed:

SQL is relational database which requires higher degree of Normalization i.e data needs to be broken down into several small logical tables to avoid data redundancy and duplication. Normalization helps manage data in an efficient way, but the complexity of spanning several related tables involved with normalization hampers the performance of data processing in relational databases using SQL.

On the other hand, in NoSQL Databases such as Couchbase, Cassandra, and MongoDB, data is stored in the form of flat collections where this data is duplicated repeatedly and a single piece of data is hardly ever partitioned off but rather it is stored in the form of an entity. Hence, reading or writing operations to a single entity have become easier and faster.

3. Flexibility:

Relational and NoSQL data models are very different. The relational model takes data and separates it into many interrelated tables that contain rows and columns. These tables reference each other through foreign keys that are stored in columns as well. When a user needs to run a query on a set of data, the desired information needs to be collected from many tables – often hundreds in today’s enterprise applications – and combined before it can be provided to the application.

Similarly, when writing data, the write needs to be coordinated and performed on many tables. When data is relatively low-volume, and when it is flowing into a database at a low velocity, a relational database is usually able to capture and store the information. But today’s applications are often built on the expectation that massive volumes of data can be written (and read) at speeds near real-time. NoSQL databases have a very different model. At the core, NoSQL databases are really “NoREL,” or non-relational, meaning they do not rely on tables and the links between tables in order to store and organize information.

4. For the type of data to be stored:

SQL databases are not much good fit for hierarchical data storage. NoSQL database are comparatively better for the hierarchical data storage as it follows the key-value pair way of storing data content which is similar to JSON data. NoSQL database are mostly preferred for large data set (i.e. for big data). Hbase is one of the examples for this.

5. Rapid Development:

NoSQL databases tend to be less complex and considerably simpler to deploy than SQL. It is easy to change how data is stored or the queries you're running in NoSQL databases. Massive changes to data can be accomplished with simple refactoring and batch processing rather than complex migration scripts and outages. It's even easier to take nodes in a cluster offline for changes and add them back into a cluster as replication features will take care of syncing up data and propagating the new data design out to the other servers in a cluster.

6. Supports JSON:

Several years ago many SQL systems added XML document support. Now, as JSON becomes a popular data interchange format, SQL vendors are adding JSON-type support as well. There are good arguments for structured data type support given today's agile programming processes and the uptime requirements of web-exposed infrastructure. Oracle 12c, PostgreSQL 9.2, VoltDB and others support JSON – often with performance benchmarks superior to “native” JSON NoSQL stores.

SQL will continue to win market share and will continue to see new investment and implementation. NoSQL Databases offering proprietary query languages or simple key-value semantics without deeper technical differentiation are in a challenging position. Modern SQL systems match or exceed their scalability while supporting richer query semantics, established and trained user bases, broad eco-system integration and deep enterprise adoption.

7. Scalability:

The most beneficial aspect of NoSQL databases like HBase for Hadoop, MongoDB, Couchbase and 10Gen's is - the ease of scalability to handle huge volumes of data. For instance, if you operate an eCommerce website similar to Amazon and you happen to be an overnight success - you will have tons of customers visiting your website. Under such circumstances, if you are using a relational database, i.e., SQL, you will have to meticulously replicate and repartition the database so as to fulfill the increasing demand of the customers.

8. For DB types:

On a high-level, we can classify SQL databases as either open-source or close-sourced from commercial vendors. NoSQL databases can be classified on the basis of way of storing data as graph databases, key-value store databases, document store databases, column store database and XML databases.

9. Data recovery:

When it comes to data recovery especially during natural crisis in big data application nosql database are easy to recover. As you know NOSql is unstructured database and data is stored in document form.

10. Data security:

Compare to SQL database, NOSQL does not provide security which is one of the main problem arise in big data application.

V. CONCLUSION

The main aim of this research paper is to evaluate which database is better for big data. Developers want a very flexible database that easily accommodates new data types and isn't disrupted by content structure changes from third-party data providers. Much of the new data is unstructured and semi-structured, so developers also need a database that is capable of efficiently storing it. Unfortunately, the rigidly defined, schema-based approach used by relational databases makes it impossible to quickly incorporate new types of data, and is a poor fit for unstructured and semi-structured data. NoSQL provides a data model that maps better to these needs.

VI. SUGGESTION

As it has been cleared compare to sql database, nosql database is a good way for big data application but queries in NoSql is not properly implemented and are not standardized. So queries must be properly implemented using some complier.

VII. ACKNOWLEDGEMENT

We thank our college IMCOST who provided insight and expertise that greatly assisted the research .We would like to express special thanks of gratitude to Prof. Trupti Deshmukh and Prof. Sunaina Raina who helped us in doing this research. We would also like to express special thanks of gratitude to all teaching and non teaching staff who gave us the golden opportunity, which also helped us in doing a lot of research and we came to know about so many new things.

We would also like to thank our parents and friends who helped us a lot in finalizing this research within the limited time frame.

REFERENCES

- [1] International Journal of Advanced Research in Computer Science and Software Engineering, 'SQL and NoSQL Databases' by Vatika Sharma, Meenu Dave.
- [2] SQL vs NoSQL Databases Differences Explained with few Examples DB by LUKE P. ISSAC on JANUARY 14,2014
- [3] <http://searchdatamanagement.techtarget.com>
- [4] International Journal of Applied Information Systems (IJ AIS), 'Types of NOSQL Databases and its Comparison with Relational Databases' by Ameya Nayak, Anil Poriya, Dikshay Poojary.
- [5] <http://www.thewindowsclub.com/difference-sql-nosql-comparision>
- [6] <http://www.digitalocean.com/community/tutorials/und-erstanding-sql-and-nosql-databases-and-different-database-models>
- [7] International Journal of Science & Engineering Research, 'Modeling and Querying Data in MongoDB' by Rupali Arora, Rinkle Rani Agarwal