Big Data and modern technologies for processing information

By: Andrei Petruk *Software Development student at GMIT*

Abstract—In the recent years, data generated by multiple types of stakeholders has been growing exponentially, computer scientists call this data the Big data. This data needs to be stored in an efficient way. Traditional relational databases are becoming less effective in meeting the needs of Big data. In this research, we will analyze the main architectural solutions and alternatives for data storage such as NoSql.

Index Terms—Big Data, NoSql, RDBMS

1 Introduction

AROUND 80-90% of the distinctive types of data in this cutting-edge time has been worked on consistently and successfully created a couple of years ago. We make rough data, advanced pictures and recordings, exchange data, and that's only the tip of the iceberg. In a situation like this, associations require an online arrangement that incorporates a significant introduction of highlights such as UIs, coordinated effort, and secure access, with concentrated and additionally massively adaptable data stockpiling as the back end, creating various sorts of substance, for example, Documents, Audio, Video, Images, Metadata in colossal sum.[1] There are three basic requirements for heavily loaded applications:

Lots of data: the largest of the web-applications process data volumes at orders greater than those anticipated for relational database management.

A huge number of users: numbered in millions, access to systems simultaneously and constantly.

Complex data: as a rule, these applications are not simple processing of tabular data, which can be found in many commercial and business applications. Relational database technologies that dominated the IT industry since 1980, began to show their weaknesses in the transition to web scales in these three aspects, so a growing number of people began to look for an alternative. Is NoSql suitable alternative to a relational database? We will access and answer this question.[2]

2 Big Data

Currently, the term Big data not only means large volumes of structured or unstructured data, how to store them, sample, analyze, but also their properties, which are described as five V - volume, velocity, variety, value, veracity [3]. Consider the quantitative examples of some of the parameters:

Volume - 2.3 trillion gigabytes of information are generated every day. Submitted scientists from IBM, by 2020 will be generated 40 zettabytes of information.

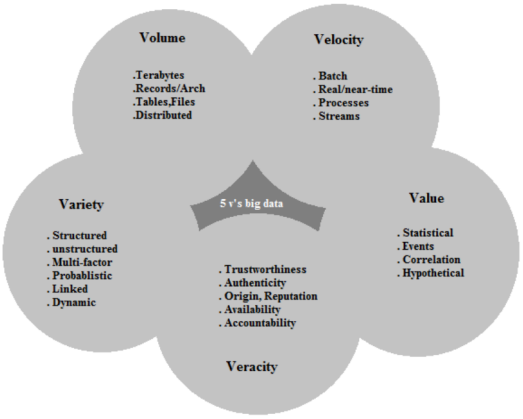
Velocity - in one trading session, the New York Stock Exchange generates about one a terabyte of information about the auction, modern cars contain about 100 sensors, and the number of network connections in computer networks can reach about 19 billion.

Variety - in 2011, the volume of all health data reached 161 billion gigabytes, every month on the youtube site you can see over 4 billion hours of video, and 200 million active users of the Twitter service send about 400 million every day

Value: Which tends to the requirement for valuation of undertaking information? It is the most vital v in Big data. Value is a primary buzz for Big data since it is imperative for organizations, IT foundation framework to store a substantial measure of values in database.

Veracity: The expansion in the scope of qualities ordinary of an extensive informational index. When we manage high volume, velocity and variety of information, the majority of the information is not going 100% right, there will be grimy infor-

mation. Big Data and examination advances work with these sorts of information. [3]



3 Relational Database

At first, information was put away in files. In any case, as the measure of data expanded, it was most certainly disadvantageous to get to the information utilizing records. It was a moderate and wasteful process. As the measure of information developed, it was exceptionally hard to keep up the information and get any record. Various levelled and Network databases were outlined as capacity systems yet they did not give a standard strategy to get to the data. With the need to oversee information and the want for a standard strategy to get to information, SQL appeared.

3.1 ACID

ACID is essential only when framework is kind banking, finance, safety systems and so on which can be overhead for frameworks that necessities to share data immensely like Amazon, Amazon etc. Principle Focus of RDBMS is on ACID properties. Atomicity: Each transaction is atomic. If one part of it fails, the entire transaction fails (and is rolled back)

Consistency: Every transaction is liable to a reliable arrangement of guidelines

Isolation: No transaction ought to meddle with another transaction

Durability: Once a transaction is submitted, it stays conferred [5]

3.2 Impediments of RDBMS to support big data

To start with, the data size has expanded massively to the scope of petabytes- one petabyte = 1,024 terabytes. RDBMS was discovered is being tested to deal with such enormous information volumes. To address this, RDBMS included more processing power (or CPUs) or more memory to the database administration framework to scale up vertically. Second, most of the data arrive in a semistructured or, on the other hand, the unstructured configuration from online networking, video, sound emails, and messages. Nonetheless, the second issue related to unstructured information is outside the domain of relational databases on the grounds that RDBMS can't sort unstructured information. They're composed and organized to suit organized data, for example, weblog sensor and finance related information. Additionally, big data is produced at a high speed. RDBMS must be in high speed since it's intended for consistent information maintenance rather than fast growth of the data. Regardless of the possibilities and storage of big data that RDBMS provides, it will end up being extremely costly. Thus, the powerlessness of RDBMS to deal with big data, has prompted the development of new technologies: NoSQL data stores.[6]

4 NoSQL DATABASE

The ascent of Big Data made an interest for a horizontally adaptable Data Management System. This prompted advancement of various types of Database Management Systems, where all things considered go under NoSQL. NoSQL Databases are comprehensively separated into following sorts: Document, Key-value, Graph, Native object, Table type, Native XML and Hybrid Databases. All RDBMS databases depend on the same model, while, each of the NoSQL database takes a distinctive model. NoSQL moves far from the robust institutionalized type of SQL database and empowers less complex information capacity arrangements. Therefore a NoSQL database is enhanced for this particular application.[7]

4.1 NOSQL DATABASE TYPES

4.1.1 Key-Value Storage

Key-Value Storages are straightforward and simple NoSQL frameworks, for example Redis, which are fundamentally a truly favourable hash table. You have an esteem you need to get later, so you allocate it a key and place it into the database, you can just inquire a single object at any given moment and just by a single key.

4.1.2 Document Storage

Regularly, these are objects with a various levelled structure, for example, XML documents, JSON records, and some other kinds of tree structures, yet the qualities of various nodes on the tree can be indexed. They have a lot of speed with respect to customary push construct SQL databases in light of query in light of the fact that they give up execution on joining.

4.1.3 Columnar Storage

These store the information in columns instead of lines, so refreshing and adding is costly Most questions are modest in light of the fact that each segment is certainly recorded. However, on the off chance that your inquiry cannot utilize an index, you are in no better shape with a Columnar Store instead of a standard SQL database.

4.1.4 Graph Storage

Diagram Databases (neo4j) make joins as shoddy as could be expected under the circumstances, on the grounds that even a straightforward row query would require many joins to recover. Tables can sort query would be slower than a standard SQL database in view of the greater part of the additional joins to retrieve the information. [8]

4.2 Examples of NoSQL

Key-Value Storage: MUMPS, CouchDB, FoundationDB, Redis, Aerospike, Dynamo, MemcacheDB, Riak, OrientDB, Fair Com c-treeACE, Redis

DocumentStorage: Apache CouchDB, MarkLogic, OrientDB, Clusterpoint, Couchbase, MongoDB,

Columnar Storage: Vertica, Cassandra, Hbase, Accumulo, Druid,

Graph Storage: Neo4J, OrientDB, Stardog, Allegro, InfiniteGraph, Virtuoso

4.3 ADVANTAGES OF NOSQL

NoSQL databases are very versatile, dependant, have a basic data model and an amazingly exposed query language, no system for taking care of consistency, trustworthiness among data and no help for security at the database level. A standout amongst the most vital points of interest of NoSQL databases is that the databases can handle unstructured information. Unstructured data can be word reports, messages, sound, video, or even interpersonal social network information. Too, NoSQL databases tend to scale extremely well on commodity equipment. Some even claim that NoSQL databases empower better execution, which is urgent for organizations with a lot of data. To empower quicker execution, NoSQL databases ordinarily don't cling to ACID ((atomicity, consistency, isolation, durability) restrictions that are utilized as a part of relational databases. While this is recorded as a star for NoSQL as far as execution and processing time; we take note of that this likewise has unfortunate outcomes that will be tended to later. A case of NoSQL database's execution is Facebook's usage (Cassandra) that is equipped for dealing with more than 100 million clients persistently. [9]

NoSQL designers cherish the effortlessness of creating and conveying with open-source usage, for example, MongoDB or Couch DB. The plan of a database is insignificant and a document can contain any sort of data if it stays open by its key- value or archive number. For software engineers, NoSQL APIs to query and update data are less complex to structure than the regularly complex query required in conventional SQL RDBMS. Maybe the most imperative viewpoint is the speed with which designers can go from an idea to a creation. Usage like MongoDB is intended for snappy setup to enable designers to understand their confirmation of-idea in a similar short period of time. [10] Maintenance after creation is likewise kept to a base and is generally reliant on the unwavering quality of the cloud servers in which information is stored on.

4.4 CHALLENGES OF NOSQL

NoSQL is still in its early stages. There is still far to go with the NoSQL so that it can plainly and lavishly wind up utilitarian and stable framework. As a result, since there is so much work left to be done, there is less propelled aptitude in this field. It provides with BASE properties, however, it isn't as dependable as the ACID properties given by SQL databases. ACID properties of transactions are highly significant in various cases, for example, the financial sector.[11]

5 How NoSQL is superior to SQL for Big Data applications

The information in Big Data applications generally differs. The data is gathered from various sources like social media, mobiles, and so forth. The information can be close to home data of the client,

area information, machine information, sensor produced information, and so on. Handling such information in a versatile and adaptable manner is of most significance. Scaling in SQL frameworks implies burning through cash on costly equipment at a solitary hub. This vertical scaling isn't impactful and furthermore, is a conservative approach. NoSQL being on a scalable plane level can be effectively used to execute Big Data applications. Scalability in NoSQL is as simple as including a server node into the framework [11]. In this way, the heap on the framework is shared between the nodes. Adaptability is intrinsic in NoSQL databases as it doesn't have to be limited to a specific blueprint unlike Relational Databases.

6 WHY THERE IS ROOM FOR BOTH

The usage of only NoSQL isn't the right answer for all information stockpiling issues since some organizations have much more extraordinary needs than others with regards to how they utilize their database and what it is utilized for. The well known social organizing website Facebook utilizes a blend of NoSQL and MySQL to run its online system. Facebook has a huge measure of clients that go to their site each day. Facebook should have the capacity to store the majority of the data coming in at a quick and dependable rate, they utilize a blend of NoSQL and MySQL that they have been altered to fit their necessities.

NoSQL is an awesome device for use in an online web-based business stage. For instance, stores like Amazon and Barnes and Noble need an exceptionally dependable, ultra-adaptable key/value database. This is a direct result of the high volume of transfer speed coming through their servers. NoSQL isn't just constrained to online business as different organizations can profit by NoSQL and since its capacity is to relentlessly advance on various stages. A few organizations have just exchanged SQL's rich usefulness for these new choices that let them make, work with, and oversee vast data sets. A major explanation behind this development, named NoSQL, is the diverse usage of web, undertaking, and distributed computing applications that have diverse prerequisites of their databases. For distributed computing and high-volume sites, for example, Twitter, eBay, Amazon, what's more, Facebook, adaptability and high accessibility are fundamental.

Utilizing entirely SQL or NoSQL for all business rehearses isn't a choice. Each business has an alternate requirement for how they plan to utilize a database and the necessities rely upon what a business needs out of its database. The SQL and NoSQL models both have their own particular arrangement of upsides and downsides that every business needs to distinguish, and after that choose which one is better for their organization or if they should utilize a mix of both SQL and NoSQL. Two primary focuses organizations should look at when contrasting SQL with NoSQL is cost and execution. For a considerable lot of the present superior workloads, the absence of execution of SQL can cost an organization a disastrous hit in income. With the testing economy today it is anything but difficult to see why a few organizations might not have any desire to set up new frameworks and how rising organizations would settle on new NoSQL setups from a conservative viewpoint.[12]

7 CONCLUSION

The big data blast is causing companies both substantial and little to look for a superior approach to store, oversee and break down vast unstructured data sets for the upper hand. With all the above advantages, NoSQL can be an intense improvement over RDBMS for organizations hoping to accomplish more with Big Data going ahead. Consolidating the qualities of both NoSQL and RDBMS is additionally a successful approach. As it is with any new innovation. [12]

REFERENCES

[1] Rakesh, K. Shilpi, C. Somya, B Journal of Advanced Database Management Systems ISSN: 2393-8730 (online) Volume 2, Issue 2 www.stmjournals.com Available at: https://www.researchgate.net/profile/Rakesh_Kumar175/publication/280622043_Effective_Way_to_Handling_Big_Data_Problems_using_NoSQL_Database_MongoDB/links/55bf4b7208aed621de123708/Effective-Way-to-Handling-Big-Data-Problems-using-NoSQL-Database-MongoDB.pdf (Accessed: 15 October 2017).

[2] Konstantin, S. From SQL to NoSQL and back [Journal] Available at: http://www.osp.ru/os/2012/02/13014127/ (Accessed: 17 October 2017).

[3] Doug, L. Application Delivery Strategies. Available at: https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf (Accessed: 18 October 2017)

[4] Guru, N. Technologies to Handle Big Data. Available at: http://www.sbsstc.ac.in/icccs2014/Papers/Paper2.pdf (Accessed: 21 October 2017).

[5] Raut, A. B. International Journal of Computational Intelligence Research ISSN 0973-1873 Volume 13, Number 7 (2017), pp. 1645-1651 Available at: http://www.ripublication.com/ijcir17/ijcirv13n7_08.pdf (Accessed: 22 October 2017).

[6] Shields, A. (2014) Market Realist, RDBMS for data storage, available at: http://marketrealist.com/2014/07/traditional-databasesystems-fail-support-big-data/ (Accessed: 24 October 2017)

[7] Gudivada, V. Rao, D. and Raghavan, V: "NoSQL Systems for Big Data Management, (June 2014), Conference: 2014 IEEE World Congress on Services, At Anchorage, Alaska (Accessed: 25 October 2017)

[8] Kumar et al. Apache Hadoop, NoSQL and NewSQL Solutions of Big Data. International Journal of Advance Foundation and Research in Science & Engineering (IJAFRSE). Oct (Accessed: 27 October 2017)

[9] Okman, L., Gal-Oz, N., Gonen, Y., Gudes, E., and Abramov, J. Security issues in NoSQL databases trust, security and privacy in computing and communications (TrustCom), IEEE 10th International Conference, 541-547, 16-18.

[10] Hammes, D. Medero, H. Mitchell, H. Proceedings of the Southern Association for Information Systems Conference, Macon, GA, USA (March 2014) Available at: http://saisconferencemgmt.org/proceedings/2014/HammesEtal.pdf (Accessed: 29 October 2017)

[11] Richards, J. Advantages and Disadvantages of NoSQL databases what you should know, Hadoop360, (September 24, 2015) Available at: http://www.hadoop360.com/blog/advantages-anddisadvantages-of-nosql-databases-what-you-should-k (Accessed: 2 November 2017)

[12] Nance, Cory; Losser, Travis; Iype, Reenu; and Harmon, Gary, "NOSQL VS RDBMS - WHY THERE IS ROOM FOR BOTH"(2013). 27 SAIS 2013Proceedings. Available at: http://aisel.aisnet.org/sais2013/27 (Accessed: 10 November 2017)