**Big Data and modern technologies for processing information**

**Abstract**

In the recent year's, data generated by multiple types of stakeholders growing exponentially, computer scientists calling 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.

**Keywords**

Big Data, NoSQL

**INTRODUCTION**

Around 80-90% of the distinctive sorts of information in this cutting-edge time has been made a previous couple of years alone and consistently, we make roughly data, advanced pictures and recordings, exchange data, and that's only the tip of the iceberg. In such situation, associations require an online arrangement that incorporates the strong introduction highlights of a entrance like UIs, coordinated effort, and secure access, with concentrated and additionally massively adaptable information stockpiling as the back

end, making out of 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 review it and answer this question.[2]

**Big Data**

Currently, the term Big data means not only large volumes structured or unstructured data, how to store, 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 (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

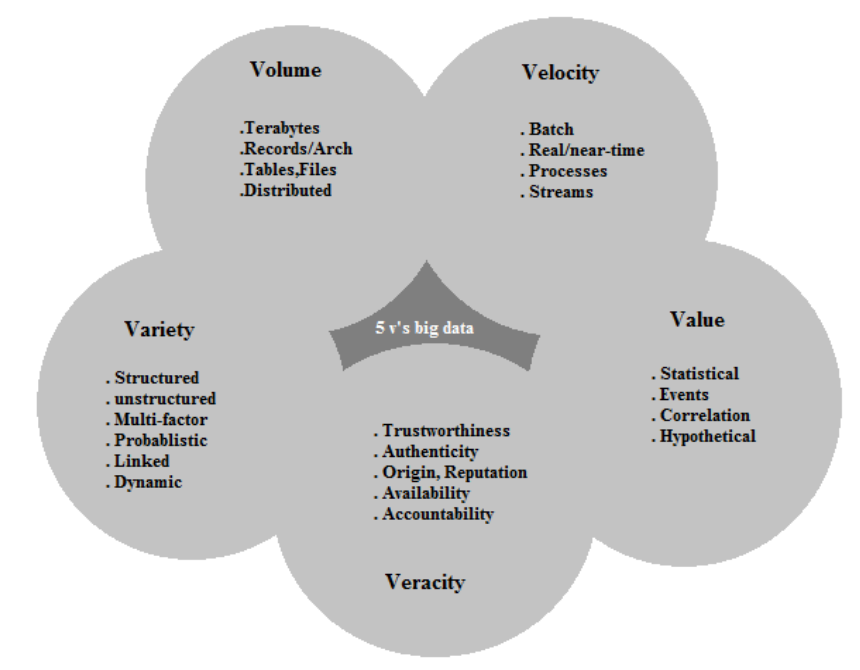
the number of network connections in computer networks can reach about 19 billion.

* 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 a most vital v in Big data. Value is 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 managing high volume, velocity and variety of information, the majority of the information is not going 100% right, there will be grimy information. Big Data and examination advances work with these sorts of information. [3]



2.0

**Relational Database**

At first, information was put away in files. In any case, as the measure of data expanded, it was most certainly not advantageous 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.

2.1

**ACID**

ACID is essential however just 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 and so on

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]

2.2

**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 discovers it testing 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 needs in high speed since it's intended for consistent information maintenance rather than fast growth of the data. Regardless of the possibility that RDBMS is utilized to deal with and store big data, 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]

3.0

**NoSQL DATABASE**

The ascent of Big Data made an interest for on a horizontally adaptable Data Management System. This prompted advancement of various types of Database Management System which 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 after 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 the particular application.[7]

3.1

**NOSQL DATABASE TYPES:**

* **Key-Value Storage**

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

* **Document Storage**

Regularly, these are objects with a various levelled structure, for example, XML documents, JSON records, and some other kind of tree structure, yet the qualities of various nodes on the tree can be indexed. They have a much 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.

* **Columnar Storage**

These store the information in columns instead of lines, so refreshing and adding are costly, be that as it may, most questions are modest in light of the fact that each segment is basically certainly recorded. In any case, on the off chance that your inquiry can not utilize an index, you are in no better shape with a Columnar Store rather than a standard SQL database.

* **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]

3.2

**Examples of NoSQL**

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

**Document Storage:** Apache CouchDB, MarkLogic, OrientDB, Clusterpoint, Couchbase, MongoDB,

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

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

3.3

**ADVANTAGES OF NOSQL**

NoSQL databases are very versatile, dependable, have a basic data model, amazingly exposed query language, no system for taking care of consistency and 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]

**RESOURCES**

[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: 20 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: 18 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: 21 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

#### [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: 21 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.