International Journal of Future Generation Communication and Networking
Vol. 12, No. 3, (2019), pp. 54 - 69

Exhaustive Comparative Study of NoSQL Database and Query Execution under Cloud Computing Environment

Dr.Mohd Ashraf, [2] Dr. Dinesh Kumar Singh, [3] Gaurav Goel
[1]*School of Technology Polytechnic, Maulana Azad National Urdu University, India*
[2, 3] *Department of Computer Science, Dr.Shankutla Mishra National Rehabilitation University, Lucknow(India)*
[1]*ashraf.saifee@gmail.com*[2]*dineshsingh025@gmail.com,*[3]*goyals24@gmail.com*

Abstract

Due to enhancement of theInternetandcloudcomputing technology, there is a need of databases to enable processing and storing the big data adequately with high performance during reading and writing the data. So the traditional relational databases are paving different challenges. Using the relational database to store data query and dynamic user applications has proven to be irrelevant, especially in massive and highly concurrent, such as search engines and SNS.In the environment of cloud, the huge data are used, the data are non-organized, the query of the data is dynamic, and these attributes raise new difficulties for the data storing and administration. Subsequently, in this situation, NoSQL database comes into the picture. Ultimately, the standard NoSQL databases have independently described and separated somebehavior to benefittrades to adoptNoSQL. It has been widely used by the big data enterprises and web companies such as Amazon, Facebook, and Google. This paper addresses the inspirations, necessities,movement, and ideas behind of NoSQL databases, and also analyzes the categories of NoSQL databases with related problems of databases importantly thefield of security and application comparing with classical relational databases.

Keywords: *NoSQL databases, Big Data,DHT (Distributed Hash Table), Hadoop, Relational Databases.*

1. INTRODUCTION:

In the era of big data, it is essential as a truestatementof the significanceofinformation with the accompanying "If your data are not developing then not atall is your business". NoSQL (Not OnlyforSQL), referred to a progressive and a mixed natural collection of non-relational data management systems; where thedatabaseis not assembled generallyon the table, and normally doesn't use SQL for data control. The NoSQL database management system is useful when it comes to a huge amount of data when the nature of the data does not need a relational model.NoSQL databases are not relational databases and distributed intended for data manipulation and enormously parallel data storage on a large scale on a large number of servers.TheyalsouseNoSQL components and dialectsassociatedwith the records. The NoSQL database system came about by true Internet organizations, such as Facebook, Amazon, and Google; which had problems in handling huge data's with standard RDBMS could not fit solution.They can increase the numerous movements, with predictive and

exploratory analysis, ETL -styledata change and no mission-basic OLTP (for example, long-term management or interrelationship connections [1].

Initially awakened by Web 2.0 applications, these systems are designed to measure to thousands or a large number of customers making redesigns and also read, instead of usual DBMS and data stores. New SQL systems which are relational databasesintended to run real-time OLTP (Online Transaction Processing), ACID (atomicity, consistency, isolation and durability) and compatible with predictable OLAP SQL-based environments of large volumes of data.Thesesystemsbreakdownoverpredictable RDBMS execution confines by utilizingNoSQL-style properties, forexample,distributed architecturesandcolumn-oriented data storing, or by utilizing advancements Massively Parallel Processing (MPP),in-memory processingorSymmetric Multiprocessing.

The general features of NoSQL data storage are:

1.1 Simple to use in traditional load-balanced clusters
1.2 Constant data (not only caches)
1.3 Scale to accessible memory
1.4 Allowsschemeand has no flat schemas relocation without downtime
1.5 Singular query systems on behalf of using the standard query language
1.6 It is the ACID within a cluster node and finally reliable throughout the cluster

2. TYPES OF NOSQL DATABASES

2.1. **Graph database** – According to graph system, databases are intendedfor the data which has associations for calling a graph and haveelements connectedwithanumber of unresolved relations. CasesincludingTitanas well as Neo4j.

2.2. **Key-Value store** – Starting this type of databases as these are some of the least complex options ofNoSQL.Databases are proposed for saving data with a lessschemaway, a key-value store.Most of the data inside consists of a value and a key indexing, and then the name.The case of these types of database includesRiak,Table Storage (ATS), Cassandra,BerkeleyDB, and DyanmoDB.

2.3. **Column store** – It is also known as a wide column store.This refers to the databases for replacing data tables forcibly as a section of data's columns, rather thanofdata's rows. Thoughstraightforwarddescription echoesas the inverse of a standard database, big column stores deal witha very versatile and elite design. Illustrations includeHBase, Big Table, and Hyper Table.

2.4. **Document database** – Development over theessential thought of key-value stores in which "documents" coverfurther complicated in a waythatholdsdata and a unique keyallottedtoeach document,

furtherutilized in document recovery. These are forced for updating, storingand managing document-oriented information and also known as semi-structured data. Examples includeMongoDB and CouchDB [1].

Key attributes are shown in this following table 1, to consider for evaluationNoSQLdatabases[2].

Table1: Comparative Study of Types of NoSQL Databases

Types of NoSQL Databases	**Flexibility**	**Complexity**	**Scalability**	**Performance**	**Functionality**
Document Store	High	Low	Mutable(High)	High	Mutable(Low)
Key valueStore	High	None	High	High	Mutable(None)
Graph Database	High	High	Mutable	Mutable	Graph Theory
Column Store	Moderate	Low	High	High	Minimal

3. PROS OF NoSQL VS RELATIONAL DATABASES

The logicbehindcommerceisgrasping a NoSQL database scenario over relational databaseshaving for all intents and purposes everything to do with the going tospecialized necessities and business area drivers. Rightwhilemaking the change, consider taking at this guide relational database to NoSQLdatabasesaimed at a walk-through of NoSQL instruction, migration andaccomplishment[3].

3.1. The Growth of Big Data

One of the maingoals to utilizeNoSQL is an accountto handle a big data task. A project of Big Data is typicallypersonified to restore the report. They are proposed for recovery,management,andstorage records-oriented info and also called as (semi) partially structured data. CaseincludesCouchDB and MongoDB[4].

3.2. ContinuousAvailability ofData

Hardware failure issueswillandcanoccur, fortunately,NoSQL database scenarioworkwith the distributed architecture,hencethere is an implied together withdataandcapacity and there is not the only point of failure. On the off adventitious that one or added database servers, or ‘nodes’ decline, the alternative nodes in the system are able to continue with processesbeyond data misfortunate, accordinglyindicatingabsolute adaptability to internal breakdown.In this sense,

NoSQL database scenarios are able to provide continuous availabilityeither in distinctregions, transversely across data centers and cloud. Regards to this point,NoSQL database can provide great achievement at an enormousscale, which never falls, and which ismassivelyuseful as the system upgrades or alterationscan be createdwithout having to disconnect the database.

3.3. Real Location Independence

The terminology "location independence" refers to the strength to write-read in touch with a database regardless of where such operation I/O appears physically and to have some write performanceraised out from that position, so theavailability is confirmed to machines and clients on various websites.So this is a complex function to designa relational database.Somemethods can be used, for instance, Slave/Mastermodels and database sharing can occasionallyfit the requirement for position free read actions, and although the writing of data all over the place is an alternate matter, particularly in huge data volumes.Other situations in which the independence of the location is again are countless to contain customer services as the dissimilar geography and it is also necessary to keep local data on those websites to access quickly.

3.4. Modern Capabilities for Transaction

The exchanging idea gives off an impression of being changed in the webera, and it has been demonstrated ACID transactions are no longer required in database based systems. In the beginning,it comes flushed, this affirmation sounds extreme, as an uprightness of value-based is virtually normal for every frame of information – mainly with those dataprerequisitesthat demand accuracy as well as safety. Anyway, what this refers to is not the endangerment of the data, but instead, the new advanced application ensures transactional consistency transversely relative to broad distribution systems.The "C" in ACID refers to the consistency of data inRDBMSthat is applied by means ofreferential integrity or foreign key constraints. This consistency type is useless inthemanagement system of dynamic data. By contrast, the "consistency" concerns to NoSQL databases lie in the CAP theorem, which means fast or possible data consistency across all nodes involved in a distributed database.The data is still protected and makes the AID bit ACID definition of Relational Database Management System;but,consistency is maintained variously to give the design and nature of the system.

3.5. Flexible Data Models

Only oneactualcauseoforganization stepof a NoSQL database system from a relational database management system is the more reliable data model that is available mostly inNoSQL database. NoSQL data model:usuallyreferred as schema-less, can support a significant number ofcases for utilization and others that are not well suitablein a Relational Database Management System.NoSQL database may recognizewide-ranging of information– unstructured, semi-structured and structured– significantly more efficiently than a relational database thatdepends on a predefined schema.NoSQL data model show effortlessly tackles

such circumstances and conveys quickactivities for both the reading and writingtasks [5].

3.6. Better Architecture

Another motivation to utilize a NoSQLdatabase is therequirementofthe most appropriate structure for thespecific applications. It is actuallyimportant that organizations receive a NoSQL platform that permits them to maintainahugedatawith regards to their applications. A few, however, not all, NoSQL solutions deliver up-to-date designs that can cope with the kind of applications that need a high degree of data sharing, scalability, and constant accessibility.

3.7. Analytics and Business Intelligence

The key driver to execute a NoSQL database scenario is the abilityto extract the data that isassembled in order to conclude experiences that places business in a superior hand factor. The separation of business intelligencesignificant fromalarge volume of data is an exceptionally problematic undertaking to accomplish with traditional relational database systems. Cutting edge NoSQL database systems not only provide storage and data management of commercial application additionally takeintegrated data analytics that bearsmomentary comprehension of complex data sets and facilitate flexible decision making.

4. DISADVANTAGES OF NoSQL DATABASES OVER RELATIONAL DATABASES

4.1. Less mature: RDBMS is very older than NoSQLdatabase.The first relational database came inthemarket about 32 years ago. Supporters of NoSQL may present this as a disadvantage citing because the ageis an indicator of elimination with the advancement of years. Relational databases have matured to become richly stable and functional system. In distinction, most of the NoSQL databases are just under the initial stage of development or deployment, and there are lots of aspects that also no longer being developed.It’s a pleasing imaginative and prescient for a developer to be tentative on the chopping fringe of technology, but attention has got to be taken to preclude any horrible penalties [6].

4.2. Less support: The traders want to satisfy that they got support in a timely manner whenever it is needed. All the RDBMS vendors have put lots of effort to be certain that such tasks and enterprises are on hand for 24 hours support through remote services of database administration;these have the skills to maintain many of the RDBMS.AllNoSQLdatabasesare to be open source, with just one or two companies dealing with aid for these. Most of them have been processed ina way of smaller startups which lack the assets to financialaid on a worldwide scale, and likewise, the convincement that settled RDBMS companies like IBM, Microsoft, and Oracle.

4.3. Business intellect and analytics: NoSQLdatabaseis planned in such an approach to remembering for Web 2.0 web applications. All things considered, the vast majority of the fundamentals of NoSQL meet these needs. Where the requirements of data stretch out past the principals'insert, read, update and delete'series of a distinctive web application. These databases reward a few elements of analysis and query adhoc.Simple queries need a bit of programming expertise and the trade intelligence tools equivalent to many businesses depend don't offer connectivity to NoSQLdatabase.Anyway, this can be able to solve the timing issue, getting some tools like HIVE or PIGgenerated to present ad-hoc query tasks for NoSQL database.

4.4. Administration: The final aim for the NoSQL database designing was to deal a solution with the need of minimum supervision, but the actual truthin the field is dissimilar. NoSQL database demands a high degree of technical proficiency with maintenance and installation.

4.5. No Advance Expertise: Considering the fact thatNoSQLdatabaseis new in the business sector and allNoSQLdevelopers are still finding out the ropes distinct Relational DBMS, which have a large number of professional designers throughout the business sector and in each field of exchange. After some time, this circumstance will resolve itself, according to the time; it is simpler to searcha trained RDBMS than a NoSQLexpert. Any sector that requiresperformingofNoSQLproperties needs an appropriate concentration, bearing in mind the above obstacles and understanding the advantages which NoSQL databases facilitate.

5. SQL VERSUS NOSQL: HIGH-LEVEL DIFFERENCES [7]

Table 2: Comparative Study of SQL and No SQL database

Metrics	**SQL Databases**	**NoSQL Databases**
Nature	Primarily well-known as Relational Database (RDBMS)	Initially well-known as distributed or non-relational database
Types[14]	One type (SQL database) with minor varieties	Wide range of types include: key-value store, record database, graph database and huge-column store
Development	In the 1970s, developed to maintain first influx for data storage purpose	In the late 2000s, developed to manipulate constraints of SQL database, principally adaptability, multi-geared up information, geo-distribution and agile progress sprints
Storage	Separate files (as "employees") are stored in rows of tables, through each columns storing a	Varies depended on databases form. (e.g. the key-value stores

Design For Data[15]	special piece of information about that record (such as "manager" "date hired," so forth.), very similar spreadsheets. Associated knowledge is stored in a unique table, then after that combined collectively when more intricate queries are achieved. For illustration, "places of work" may be saved in one table, and "workers" in a further. When a client needs to find the works handle of an employee, the database engines join the "employee" and "offices" tables collectively to acquire all of the understanding integrals.	perform similarly to SQL database),however, it has simply two columns ("key" and "value"), with more unpredictable data typically stored as BLOBs inside the "value" column. Document database casts off the table-and-row model altogether, storing all vital information collectively in single "document" in JSON, XML, or other structure, which will settle values hierarchically.
Schemas	Data types and Structure are altered prematurely. To save data into a new data article, the whole database has to be manipulated, during that time the database needs to be taken offline.	Regularly innovative, with some imposing data authentication guidelines. Functions can comprise new tags on the fly, and in contrast to SQL table rows, divergent knowledge may also be saved together as indispensable. For a couple of databases (e.g., large-column stores), that is more challenging to add new fields dynamically.
Properties	This accentuates on ACID properties (Atomicity, Consistency, Isolation and Durability)	This takes after the Brewers CAP theorem (Consistency, Availability, and Partition acceptance)
Scaling	Vertically, implies a single server ought to be made a step by step process so that you can deal with multiplied demand. It is imaginable to spread SQL databases over countless servers, but colossal additional engineering is required in most cases, and core relational features, for example, JOINs, referential integrity and transactions are in general misplaced.	Horizontally, intended to comprise potential, a database admin can easily incorporate external commodity servers or cloud simulators. The database spreads information on a routine basis over the servers.
Progress Model	Mixture of locked source (e.g. Oracle Database) and open-supply (e.g. Postures, MySQL)	It is available free of cost.
Manipulation of Data[16]	Particular language utilizing Insert, Select and Update statement, e.g. SELECT fields FROM	Utilizing with object-oriented Application Programming

	table WHERE…	Interfaces
Consistent	May also be designed for robust consistency	Clear dependency on products. Some facilitate solid consistency e.g. MongoDB by tuneable consistency for peruses while other presents subsequent consistency e.g. Cassandra.
Scalability	SQL database is vertically scalable. You can oversee high capacity by including the SSD, RAM, CPU etc. on only one server	You could simply incorporate couple of extra servers effectively for your NoSQL database infrastructure to grip the enormous traffic
Transactional Supports	Surely, updates may also be organized to conclude completely or not	Indefinite situations and at distinct phases (e.g. Database vs. document)level
Support	Superb backings are available for all SQL database from their merchants	Continuity to ought to depend on community support, and just confined external specialists are available to set up and set up wide scale NoSQL deployment
Examples	Microsoft SQL Server, Postgres, Oracle Database, and MySQL	Neo4j,MongoDB,HBase, and Cassandra

6. SELECTING THE RIGHT NOSQL DATABASE

Key concerns whileselectingNoSQL platform comprise:

6.1. **Workload diversity** – Big Data's come in all hues,sizes,andforms. Inflexibleschemes have no place at this point; to a certain extent,yourequire a more adaptableconfiguration.On the off chance that you needyour technology to appropriate your data, not the different manner as the capacity to execute transactions in real time, run analytics generally as quick, to find something you need a moment from huge data and to accomplish more with all of that data, regardless of from what that data may take.

6.2. **Scalability** – With big data you need to be capableof scalingelasticallyand quickly, wherever and whenever you need. This is applicable inallcircumstances, whether scaling over unusual servers and even to the cloud if necessary.

6.3. **Performance** – As previously described, a world wide web where microseconds or nanoseconds delay may cost your trades. Huge Data must transfer at largely high speeds regardless of the amount you expect or what workloads your database has to perform. Implementation of thebackgrounds,specificallythe applications, ought

to be great on the rundown of necessities for conveying a NoSQLplatform.

6.4. Continuous Availability – Building of the performance consideration, revenue producing 24 hours' business applications, when you depend on huge data to feed your essential, even high availability is not moresufficient. Data can by no means go down, therefore no single point of failure should be inNoSQLplatform,and as a consequence making certain functions are continually available.

6.5. Manageability – The operational complexity of a NoSQL platform must stay at least.Be certain that the developmentand administration required for each maximizesandthepreserve relocating benefits to a NoSQLplatform are acquirable[7].

6.6. Cost – This is surely an obvious purpose for creating the movement of the NoSQL platform as assembly, even one of the most issues awarded right here with relational database technique, the cost can end up prohibitively steeply-priced.DeployingNoSQL thoroughly enables for the entire above benefitswith lower operational costs.

6.7. Strong Community – This is perhaps one of the most important influences to consider as you go to a NoSQLplatform.Be certain that there's a powerful and expert field around the innovation, as this may occasionally supply an invaluable potential to the neighborhood and agencies to be able to be handling the trouble. Association with respect to the vendorsshould not just incorporate a solid backing and specialized asset accessibility, additionally steady effort to the client base. A just right nearby consumer crew and meetupswill deliver many possibilities for communicating with other individuals and groups to provide an overview of the quality of how it can work properly with the preferred platform.Most of us are already accustomed to SQL databases and have good talents of MySQL, Oracle, or other SQL database. Recently in a few years,NoSQL databasesareobtainingusuallyexpected to take care of different business issues. It is useful to grasp the distinction amongst NoSQL and SQL databases, and some of theaccessibleNoSQLdatabasescontributed around to play with[8].

7. COMPARATIVE STUDY:

A qualitative method is used to identify interesting products. Thus the study of different NoSQL products is carried out using research articles, academic papers, community forums and product documentation. A comparison of interesting NoSQL databases is carried out against desired properties for a storage solution. The comparison is listed in the table below[9].

Table 3: Comparative Study of NoSQL Databases

Criteria	**Mongo DB[10]**
Description	It is a document-oriented data store intended to scale out.
Availability	Supports high accessibility through asynchronous replication
Elasticity	New nodes can be included or expelled online
Read performance	It supports high read execution through memory-mapped files.
Scalability	It automatically distributes data and load over different servers.
Computational performance	It underpins collection through Map Reduce, rich questions and total devices. Be that as it may, map diminish occupations are moderate and ought to be executed as
Complexity	It gives a JavaScript shell to query and control data in Mongo DB server.
Support for Indexes	Supports index on any trait and conglomeration tools. It gives support for unique, compound and geospatial
High throughput for bulk inserts	Provides built-in support for mass inserts and henceforth High throughput is feasible for mass loading.

Hive[11]	**Hadoop**
It is a data warehousing framework based on top of Hadoop.	It is an open source framework that supports processing of huge datasets using a distributed
High accessibility provided by HDFS.	High accessibility provided by Hadoop Distributed File System (HDFS) and YARN
New nodes can be included or expelled dynamically.	New nodes can be included or evaluated dynamically.
It gives preferable read execution than Hadoop as a result of primary and secondary indexes over the whole data set.	It is not suited for irregular reads and low latency reads since each read operation is a scan.
It scales linearly and underpins auto-sharing.	Scales horizontally and supports auto-sharing.
Supports aggregation through SQL queries. These queries are changed into Map Reduce jobs.	Supports conglomeration through Map Reduce jobs which are keep running as batch jobs and do not giveaconstant performance.
Provides Hive Query language which is SQL-like.	It gives Map Reduce APIs in Java. It gives an interface to C++ through Hadoop Pipes and to other languages through Hadoop Streaming
Supports primary and secondary indexes.	Supports just primary index.
High throughput forthemass insert is supported.	High throughput forthemass insert is bolstered since they are executed as Map Reduce jobs.

Base[12]	**Pig**
It is a distributed column-oriented database intended to scale linearly.	It is a scripting language intended to streamline data structures and Map
High accessibility provided by HDFS and Zookeeper.	High availability provided by HDFS
At the point new nodes are included dynamically, there is no big data transfer from the old nodes to new nodes and hence the new node start serving requests	New nodes can be included online yet data will be distributed from the old nodes to the new nodes and it requires time before new nodes start serving
A row is sorted by primary key and supports low latency for sequential and random reads.	It is slower than Java Map Reduce programs and henceforth execution is slower than Hadoop.
It supports linear scalability and auto-sharing	It forms colossal volumes of data by basic programs. It scales out and
Underpins aggregation through Map Reduce Jobs.	Supports Map Reduce functionality
Provides a shell with simple commands to perform questioning and controlling data.	It gives a level of abstraction through APIs and simplifies Map-Reduce programming.
Supports primary index efficiently. Although secondary indexing is	Supports just primary index
High throughput for mass insert is upheld since they are executed as Map Reduce jobs	High throughput for mass inserts is supported.

Redis	Cassandra[13]
It is an in-memorykey-value store that is very fast.	It is a distributed storage system supposed to control big volumes of structured data
highly available	High accessibility is provided by gossip based membership protocol and failure detection module.
Adding or expelling nodes is complex and takes a lot of time since hash slots will be moved around	New nodes will also be incorporated on-line but data shall be distributed from the historic nodes to the brand new nodes and it takes time earlier than new nodes start serving requests.
It is more of a cache since it stores data in memory and hence it’s quite fast for random reads and writes.	It utilizes in-memory structure called bloom filters which are like a cache and is checked first for reading requests. Hence it provides good read performance.
Scalable however sharing has to be taken care manually.	It scales incrementally and underpins auto-sharing.
Provide quick real-time analytics.	Aid for Map lower will not be native and is slower than Hadoop. Despite the fact that it can be used together with Storm for real-time computation
Provides APIs yet no ad-hoc querying feature.	Provides Cassandra Query Language (CQL), command line interfaces and APIs.
Supports just primary index.	Supports primary and secondary indexes.
High throughput for mass inserts is supported.	High throughput for bulk inserts is supported

8. ARCHITECTURE OF NoSQL QUERY EXECUTION FLOW IN CLOUD ENVIRONMENT

a) The user generates a request which is converted in the form of a query for fetching the result.

b) Submitting this user-generatedquery using REST API call.

c) Parse this query in low-level language using the compiler and analyze it based on the request.

d) Now validate the request and create the index for this.
e) Generate the keys related to its validation and type.
f) Validate the key with the generated request and get the response in the form of the document.
g) Evaluate the document and make it in the proper format.
h) Finally returns the result in the JSON format.

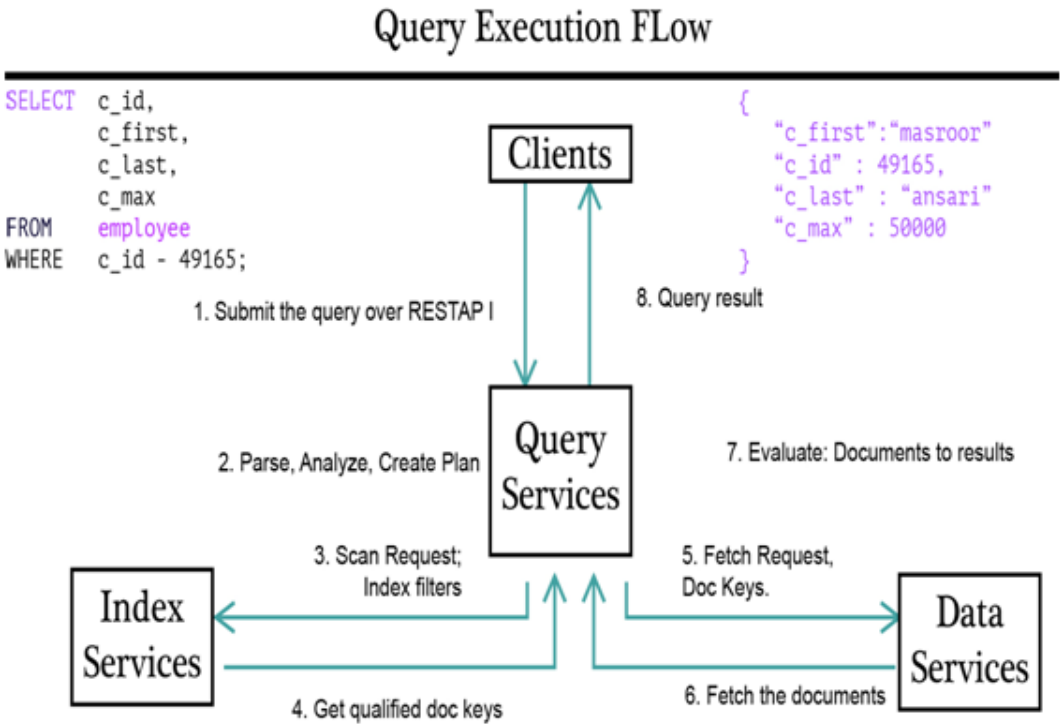

Figure 1: Architecture of NoSQL Query Execution Flow

9. CONCLUSION

With the approach of WEB2.0,thepotential user needs more and more data. As a result, the recent relational database products facing critical load issues. Countless data need to be collaborated with the database, so there is a requirement of circulated and adaptable structure. NoSQL databasebargainshigherespeciallytheconsistency of information.Technically, the content of the distributed computing unit is not designed for the relational databases with low-cost. They arewidelystraining concentration or a restricted node cluster structure.Sothe RDBMS cannot support world-class technology for information management,such as Amazon Dynamo, Google Big Table etc.The study in this paperconveyed reserving instrument, and presented the information consistency arrangement; there are numerous steady hash calculations, viz., Data Partitioning, Map Reduce Execution, tattle DHT and so on. All of them have their own features. Authors have taken the slave server as an intermediary server in the Master/Slave model and described the distributed caches corresponding to query caching in the databasesdifferent approacheshave been used tomeasuring the databases in order to minimize the database loads.In this paper we compare the NoSQL

database with different RDBMS.In future work, there is scope to lessen the hitting ration calculation on heap on local database server and proficientlt guarantee the consistency as database are overhauled.

References

[1] Grolinger, K., Higashino, W. A., Tiwari, A., &Capretz, M. A. (2013). Data management in cloud environments: NoSQL and NewSQL data stores. Journal of Cloud Computing: Advances, Systems and Applications J Cloud ComputAdvSyst Appl,2(1), 22. doi:10.1186/2192-113x-2-22.
[2] C. S., &Kriha, P. W. (n.d.). Http://webpages.uncc.edu/xwu/5160/nosqldbs.pdf.
[3] Li, Y., &Manoharan, S. (2013). A performance comparison of SQL and NoSQL databases. 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM). doi:10.1109/pacrim.2013.6625441.
[4] A. K. (2014). NoSQL Databases: New Millennium Database for Big Data, Big Users, Cloud Computing and its Security Challenges. International Journal of Research in Engineering and Technology IJRET,03(15), 403-409. doi:10.15623/ijret.2014.0315080.
[5] Relational Database Management Systems. (n.d.). Advanced Data Management For SQL, NoSQL, Cloud and Distributed Databases. doi:10.1515/9783110441413-006.
[6] Atzeni, P., Bugiotti, F., & Rossi, L. (2014). Uniform access to NoSQL systems. Information Systems,43, 117-133. doi:10.1016/j.is.2013.05.002.
[7] Leavitt, N. (2010). Will NoSQL Databases Live Up to Their Promise? Computer,43(2), 12-14. doi:10.1109/mc.2010.58
[8] M. A. (2010). "NoSQL" databases in CMS Data and Workflow Management. 13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (pp. 22-27).
[9] Deka, G. C. (n.d.). NoSQL Databases. Handbook of Research on Cloud Infrastructures for Big Data Analytics, 186-215. doi:10.4018/978-1-4666-5864-6.ch008.
[10] Han, J., E, H., Le, G., & Du, J. (2011). Survey on NoSQL database. 2011 6th International Conference on Pervasive Computing and Applications. doi:10.1109/icpca.2011.6106531.
[11] A, V. A., A, J. B., & B, P. F. (2014). Which NO SQL Database? A performance review. Open Journal of Databases (OJDB),1(2), 2014th ser.
[12] Assunsio, F., Levi, M., & Furtado, P. (2015). Comparing SQL and NoSQL approaches for clustering over big data. International Journal of Business Process Integration and Management IJBPIM,7(4), 335. doi:10.1504/ijbpim.2015.073657.
[13] Lourenço, J. R., Cabral, B., Carreiro, P., Vieira, M., & Bernardino, J. (2015). Choosing the right NoSQL database for the job: A quality attribute evaluation. Journal of Big Data,2(1). doi:10.1186/s40537-015-0025-0.
[14] Sharma, V., & Dave, M. (2012, August). SQL and NoSQL Databases :International Journal of Advanced Research in Computer Science and Software Engineering. pp 20-27,2(8).ISSN: 2277 128X

[15] Jmtauro, C., S, A., &A.b, S. (2012). Comparative Study of the New Generation, Agile, Scalable, High PerformanceNoSQL Databases. International Journal of Computer Applications IJCA,48(20), 1-4. doi:10.5120/7461-0336.
[16] Zhao, L., Sakr, S., & Liu, A. (2014). An Overview of the NoSQL World. Large Scale and Big Data Processing and Management, 287-324. doi:10.1201/b17112-10
[17] A Mongo DB white paper," Mongo DB NoSQL database", 2016, https://www.mongodb.com/nosql-explained.
[18] Thorne, D., Pettifer, S., & Marsh, J. (2004). Database (NoSQL, Quad Store, RDF Database, Relational Database, and Triple Store). Dictionary of Bioinformatics and Computational Biology. doi:10.1002/9780471650126.dob0863.
[19] Cattell, R. (2011). Scalable SQL and NoSQL data stores. ACM SIGMOD Record SIGMOD Rec.,39(4), 12. doi:10.1145/1978915.1978919.
[20] A, V. A., A, J. B., & B, P. F. (2014). Which NoSQL Database? A performance review. Open Journal of Databases (OJDB), pp 17-24 Vol1(Issue 2), ISSN 2199-3459.