Modeling and Querying NOSQL Databases

Neenamol Joseph [1], Ponnu Mathew [2], Prof. Gibin George

[1]Pursuing master's degree program in Santhigir College of Computer Science, Vazhithala, Idukki, Kerala, India
Emai:mca2022_neenamoljoseph@santhigiricollege.com

[2]Pursuing master's degree program in Santhigir College of Computer Science, Vazhithala, Idukki, Kerala, India
Email:mca2022_ponnumathew@santhigiricollege.com

[3]Assistant Professor in Santhigiri College of computer science Vazhithala, India
Email: George.gibin@santhigiricollegel.com

Abstract: ***Relational databases are providing data storage for several decades now. However for today's web and mobile applications the importance of scalability in data model cannot be over-stated. The term NoSQL widely covers all non-relational databases that provide schema-less and scalable model. NoSQL databases also termed as Internet age databases. They are currently used by Google, Amazon, Facebook and other organizations operating in the era of Web 2.0. Different types of NoSQL databases namely key-value pair, document based, column-oriented and graph based databases enable programmers to model the data closer to the format as used in their application. In this paper, modeling and querying of relational and some types of NoSQL databases are explained with the help of a case study of a news website like Slashdot.***

Keywords: json, mongodb

1.Introduction

After 1990's due to popularity of HTTP, the cost of posting and exchanging information became cheaper which led to the flooding of information on Internet. It was realized that traditional techniques of data storage will soon become stale and inefficient to handle such vast amount of unstructured and semi structured data. Not all the information generated on Web is structured, rather interactive Web has produced more semi-structured or un-structured data. All the available rich information cannot be forcefully made to fit in the tabular format of relational databases. This problem was also faced by object-oriented databases under the name "Object-Relational Impedance Mismatch" problem. This mismatch occurs when the objects are molded to fit into relational model. A large percentage of digital information floating around the world is in PDF, HTML and other types of formats which cannot be easily modeled, processed and analyzed. Amounts of data. Their data model is based on single machine architecture and was not designed to be distributed. Today all software's are developed expecting a large user-base, which was not the case in 1970s. Scalability is one of the most discussed issues today, since web applications have got enormous popularity. Scalability can is either vertically or horizontally. Vertical scalability, also known as (a.k.a.) scaling up is easier to achieve as compared to horizontal scalability a.k.a. scaling out. As the name suggests, scaling-up means adding up resources to asingle node and scaling-out means adding more nodes to a system. Horizontal scaling provides more flexibility as commodity servers or cloud instances can be utilized. Traditional databases relies on vertical scaling whereas recently evolved non-relational databases use horizontal scaling for achieving scalability. Although relational databases have matured because of their prolonged existence. Unfortunately for most of the today's software design, relational databases show their age and do not give good performance especially for large data sets and dynamic schemas. We live in a world, where domain model is constantly changing during development phase and even after deployment. These changes in requirements along with various other reasons described above, led to the development of non-relational databases known as NoSQL databases. Popularity of non-relational databases can be imagined by the fact that many universities have started teaching about the data stores their curriculum.

NoSQL is a term commonly used to cover all non-relational databases; it stands for Not only SQL or Non-SQL. There is a disagreement on this name, it does not focuses on the real meaning of non-relational database, non-ACID properties, schema-less databases, since SQL is not the obstacle as implied by the term NoSQL. The term "NoSQL" was introduced by Carlo Strozzi in 1998.it was a name for his open source relational database that did not offer a SQL interface. The term was re-introduced in October 2009 by Eric Evans for an event named no: sql (east) organized for the discussion of open source distributed databases.The name was an attempt to describe the increased number of distributed non-relational databases that emerged during the second half of the 2000's. Increasing number of players dealing with WWW started recognizing the in-efficiency of relational databases to handle huge amount of diverse data generated by the introduction of Web 2.0 applications. Google is the first to lead this movement by introducing Big Table in 2006, it followed by Amazon's Dynamo in 2007. Influenced by adoption of non-relational databases by these big Firms most of the organizations started developing their own

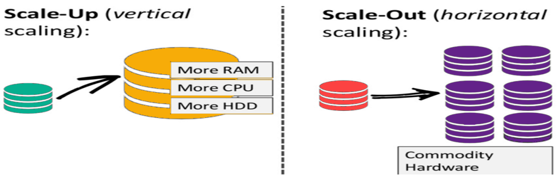
NoSQL data stores are customized according to their requirements. Most of today's popular NoSQL data stores have adopted ideas from Google's BigTable or Amazon's Dynamo. Those inspired by BigTable are divided as column-oriented or wide-table data stores and others which are descendants of Dynamo are termed as key-value based data stores. There are two other categories, document and graph-based databases. These four classes of NoSQL databases deal with different types of data and hence are suitable for different use cases. There are four attributes that are responsible for the NoSQL movement:

- **Volume** – the amount of data that requires processing and storage in the database
- **Speed** – the need to process data as soon as possible, regardless of their amount
- **Variability** – a rigid scheme and lack of dynamics in relational databases pose a problem for changes that are therefore expensive and a lot of time is lost
- **Agility** – the need for simplicity when entering and retrieving data from the database.

1.1 NoSQL Database

1.1.1 Why NoSQL Database?

The concept of NoSQL databases became deals with Internet giants like Google, Facebook, Amazon, etc. who manage with large volumes of data. The system response time became slow when we use Relational Database Management System for massive volumes of data. To resolve this problem, we could "scale up" our systems by upgrading our existing hardware of the system. This process is very expensive. The solution for this issue is to distribute the database load on multiple hosts when load increases. This process is called as "scaling out."



NoSQL database is non-relational database, so it scales out better than relational databases they are designed for web applications.

1.1.2. Brief History of NoSQL Databases

- 1998- Carlo Strozzi use the term NoSQL for his lightweight, open-source relational database
- 2000- Graph database Neo4j is launched
- 2004- Google BigTable is launched
- 2005- CouchDB is launched
- 2007- The research paper on Amazon Dynamo is released
- 2008- Facebook's open sources the Cassandra project
- 2009- The term NoSQL was reintroduced

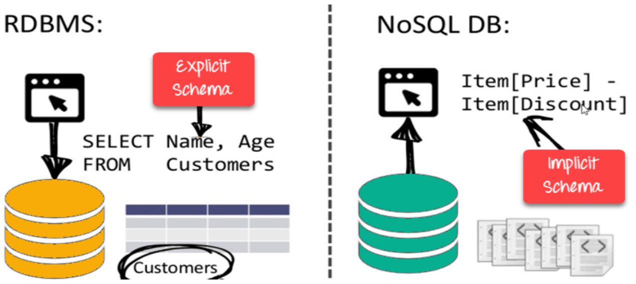
1.1.3. FEATURES OF NoSQL

Non-relational

- NoSQL databases never follow the relational model
- Never provide tables with flat fixed-column records
- Work with self-contained aggregates or BLOBs
- Doesn't require object-relational mapping and data normalization
- No complex features like query languages, query planners, referential integrity joins, ACID

Schema-free

- NoSQL databases are either schema-free or have relaxed schemas
- Do not require any sort of definition of the schema of the data
- Offers heterogeneous structures of data in the same domain

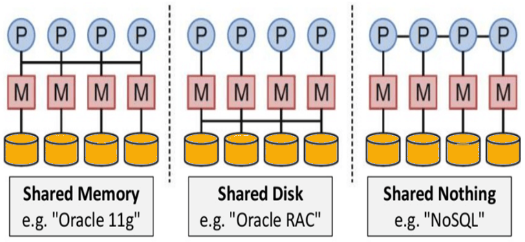


Simple API

- Offers easy to use interfaces for storage and querying data provided
- APIs allow low-level data manipulation & selection methods
- Text-based protocols mostly used with HTTP REST with JSON
- Mostly used no standard based NoSQL query language
- Web-enabled databases running as internet-facing services

Distributed

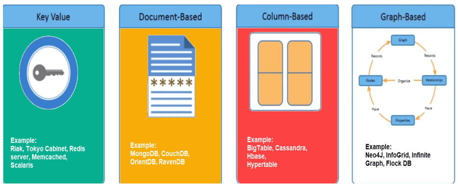
- Multiple NoSQL databases can be executed in a distributed fashion
- Offers auto-scaling and fail-over capabilities
- Often ACID concept can be sacrificed for scalability and throughput
- Mostly no synchronous replication between distributed nodes Asynchronous Multi-Master Replication, peer-to-peer, HDFS Replication
- Only providing eventual consistency
- Shared Nothing Architecture. This enables less coordination and higher distribution.



1.1.3. Modeling of NoSQL databases

There are four basic types of NoSQL systems that can be divided according to the data model. Below are their names and several prototypes (George, 2013):

- Key – value Pair
- Column-family
- Document-store
- Graph


Key Value
Example:
Riak, Tokyo Cabinet, Redis server, Memcached, Scalaris
Document-Based
*****
Example:
MongoDB, CouchDB, OrientDB, RavenDB
Column-Based
Example:
BigTable, Cassandra, Hbase, Hypertable
Graph-Based
Graph
Records
Records
Nodes
Organize
Relationships
Have
Have
Properties
Example:
Neo4J, InfoGrid, Infinite Graph, Flock DB

1. Key-Value Pair Databases

The Key-Value Pair Databases stores data in simplistic manner, but are quiet efficient and powerful model. It has a simple application programming interface (API). A key value data store allows to store data in a schema less manner. The data is a kind of data type of a programming language or it is an object. The data consists of two parts, a string which represents the key and the actual data which is to be referred as value thus creating a “key-value” pair. These are similar to hash tables, because the keys are used as the indexes, it making it faster than Relational Database Management System, Thus the data model is simple, a map or a dictionary that allows the user to request the values according to the key specified.

The modern key value data storesprefer high scalability over consistency. Hence ad-hoc querying and analytics features like joining functions and aggregate operations have been omitted. High concurrency, fast lookups and options for huge storage are provided by key-value pair data stores. One of the weaknesses of key value data sore is the lack of schema which makes it much more difficult to create custom views of the data. Key value data stores can be used in situations where you want to store a user’s session or a user’s shopping cart or to get details like favorite products. Key value data stores can be used in web-based forums, online shopping sites etc. Although key-value data stores existed for long time ago, the development of large number of recent key value data store was influenced by the introduction of Amazon’s Dynamo. Some notable key-value data stores are mentioned below.

1.1 Amazon Dynamo DB

Amazon Dynamo DB is a newly released fully managed NOSQL database service offered by Amazon that provides a fast, highly reliable and cost-effective NOSQL database service designed for internet scale applications. It is implemented using Amazon’s Dynamo model. It offers low, predictable latency. It stores data on solid state drives (SSD) instead of hard drives thus providing easier access to the data. The data is duplicated synchronously across on multiple AWS Availability Zones in an AWS Region to provide built-in high availability and data durability. It duplicates data across at multiple data centers, it providing high availability and durability under complex failure scenarios.

1.2 RIAK

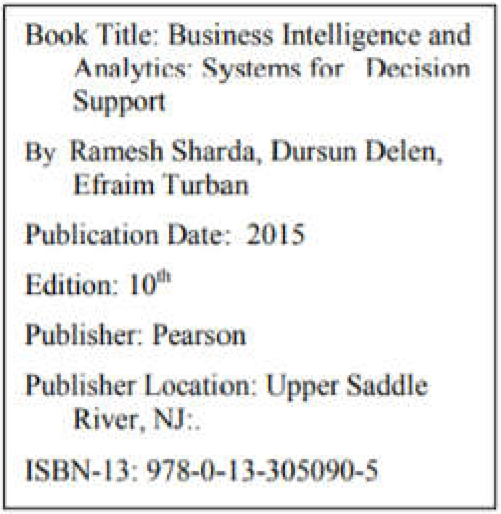
Riak is a distributed database, fault tolerant and open source database developed by Basho technologies using C, Erlang and JavaScript. It implements principles from Amazon’s Dynamo paper. It has a flexible data schema. It offers high availability, partition tolerance and persistence. Components of Riak are Riak Clients, Web machine, Protocol Buffers, Riak Replication, Riak SNMP/JMX, Riak KV, Riak Search, Riak Pipe and Riak Core. Riak should be avoided for highly centralized data storage projects with fixed, unchanging data structures. Riak is used by Mozilla, AOL and Comcast.

It can be used for following purposes

- Managing personal information of the user for social networking websites or MMORPGs (Massively Multiplayer Online Role Playing Games)
- To collect checkout or POS (Point of sales) data
- Managing Factory control and Information systems
- Building Mobile Applications on cloud etc.

Example

An example of a particular BOOK document is shown below


Book Title: Business Intelligence and Analytics: Systems for Decision Support

By Ramesh Sharda, Dursun Delen, Efraim Turban

Publication Date: 2015

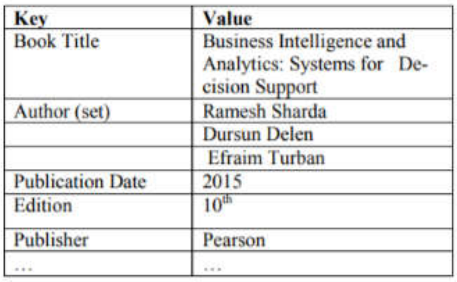
Edition: 10th

Publisher: Pearson

Publisher Location: Upper Saddle River, NJ:.

ISBN-13: 978-0-13-305090-5

Key-Value Pairs – stores information in form of matched pairs with only two columns permitted - the key (hashed key) and the value (Moniruzzaman & Hossain, 2013). The values can be simple text or complex data types such as sets of data. Data must be retrieved via an exact match on the key. The advantage of this type of NoSQL database is that new types of data about a book can easily be added to the database as new key value pairs. Examples of NoSQL databases that use Key-Value Pairs are Project Voldemort, Cache and Dynamo. In the prior book example, the book information from Figure 1 would be stored as shown in Table 1.



Key	Value
Book Title	Business Intelligence and Analytics: Systems for De-cision Support
Author (set)	Ramesh Sharda
	Dursun Delen
	Efraim Turban
Publication Date	2015
Edition	10th
Publisher	Pearson
…	…

2.Column family Databases

Column stores in NO SQL are literally hybrid row/column store not like pure relative column databases. though it shares the concept of column-by-column storage of columnar databases and columnar extensions to row-based databases, column stores don't store data in tables however store the data in massively distributed architectures. In column stores, every key is related to one or additional attributes (columns). It offers high scalability in data storage. The data that is stored on within the database is based on the sort order of the column family. Column based databases are appropriate for data mining and analytic applications, wherever the storage methodology is good for the common operations performed on the data. Some of the notable column oriented databases are mentioned below.

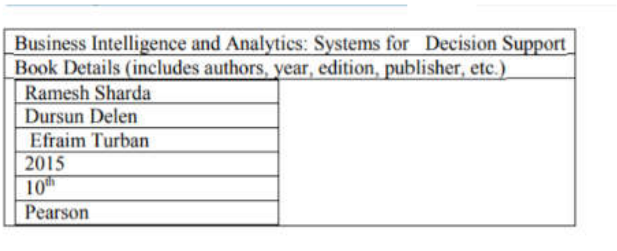
2.1 Big Table

Google's Big Table may be a compressed high performance information that was initially released on 2005 and is made on the Google File system (GFS). It had been developed by C and C++. It offers consistency, fault tolerance and persistence. it's designed to scale across thousands of machines and it is simple to add additional machines to it. the big Table implementation has 3 major components: a library that's linked into each client, one master server, and many tablet servers. tablet servers are accustomed manage a group of tablets (same as tables in RDBMS). The master server handles schema changes, performs tasks like distribution servers to tablet servers, balancing tablet server load, garbage collection etc. Big Table isn't distributed outside Google, however it is available on Google app engine. huge Table is used by variety of Google applications like Gmail, YouTube and Google Earth.

2.2 Cassandra

Cassandra was developed by Apache Software Foundations and was released in 2008. it had been developed by Java. it' is based on Amazon's Dynamo model and Google's Big table. therefore it involves ideas of each key-value stores and column stores. It offers feature like high accessibility, partition tolerance, persistence, high scalability etc. it's a dynamic schema. It are often used for a range of applications like social networking websites, banking and finance, real time knowledge analytics, on-line retail etc. Cassandra is being used by Adobe, Digg, eBay, Twitter etc. The disadvantage of Cassandra is that reads are relatively slower than writes. Example

An example of how data is stored in a column-oriented NoSQL database.



Business Intelligence and Analytics: Systems for Decision Support
Book Details (includes authors, year, edition, publisher, etc.)
Ramesh Sharda
Dursun Delen
Efraim Turban
2015
10th
Pearson

3.Document oriented Databases

Document oriented Databases area unit refers to knowledge bases that store their data within the variety of documents. It stores provide nice choices for potency and horizontal quantifiability. Among a document-oriented info, documents area unit terribly just like records in relative databases, however since they're less schematic, they're conjointly a lot of versatile. The documents area unit on the market in customary formats likes XML, PDF, JSON, etc. In relative databases, a record among identical info can have identical knowledge fields and therefore the unused knowledge fields are unbroken empty, however within the case of document stores, every document could have similar and totally different knowledge. Info documents area unit addressed employing a distinctive key that represents that document. These keys may be a straightforward string or a string that refers to a URI or a path. Document Store they're slightly a lot of advanced than key-value stores, as they permit key-value pairs to be embedded in a very document conjointly called key-document pairs. Document-oriented knowledgebase ought to be used for applications wherever data don't ought to be keep in a very table with uniform size fields, however wherever the info ought to be keep as a document with special characteristics Document Stores ought to be avoided. If the info goes to own heaps of relationships and standardization. They'll be used for content management systems, diary software package, etc. Some document knowledge stores area unit listed below.

MongoDB

MongoDB was at the start free in 2009. it had been developed victimization C++. it's a high performance and economical info. It provides options like consistency fault tolerance, persistence. MongoDB provides extra options like aggregation, adhoc queries, indexing, automobile sharing etc. In MongoDB the documents area unit chiefly keep in BSON (Binary JSON) format. BSON documents contain Associate in Nursing ordered list of components consisting of field name, kind and price. BSON is economical each in space for storing and scan speed when put next to JSON. MongoDB uses GridFS as a specification for storing giant files. MongoDB is well matched for applications like content management systems, archiving, real time analytics etc. MongoDB is presently being employed by MTV networks, Foursquare, The Guardian etc. it's additionally being employed in comes like CERN‟s LHC, UIDAI Adhere that is India's distinctive identification project. The disadvantages area unit that it is unreliable and categorization takes up ton of ram.

CouchDB

CouchDB was developed by Apache software foundation.it was at the start discharged in 2005. it absolutely was developed victimization C++. It uses JSON documents to store information and provides relaxing communications protocol API to form and update information documents. It provides JavaScript as a question language. It provides a in-built net application referred to as discoverer which may be used for administration. it's extremely obtainable, fault tolerant and protracted. It implements Multi-Version Concurrency management (MVCC) so providing coinciding access to users. CouchDB has nice replication and synchronization capabilities. It is used for applications involving sometimes dynamic information on those pre-defined queries got to be used. It is utilized in cases wherever network affiliation could or might not be obtainable; however the applying should keep it up operating, like within the case of mobile device based mostly applications. It is used for

CRM (Customer Relationship Management) and CMS systems. CouchDB is being employed by websites like LotsOfWords.com and friendpaste.com additionally by Facebook apps like Horoscope, Birthday salutation Cards etc. a number of the drawbacks of CouchDBar temporary views in CouchDB on giant datasets are extremely slow, not sensible at coping with relative information, no support for ad-hoc queries.

Example

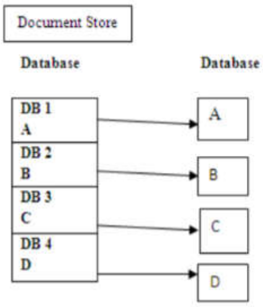


Figure 5: Structure of document stores database

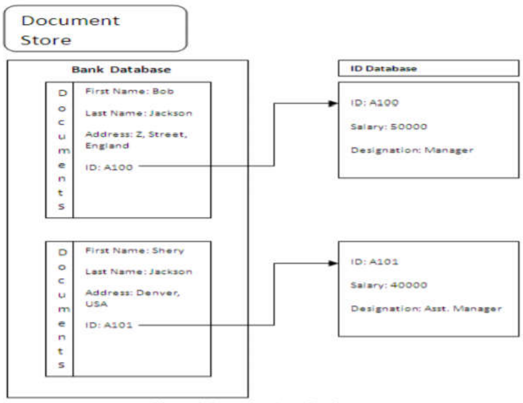


Figure 6: Document Store Databases

4.Graph Databases

Graph databases are databases that store data within the kind of a graph. The graph consists of nodes and edges, wherever nodes represent the objects and edges represent the relationship between the objects. The graph consists of properties associated with nodes. It uses a method known as index free adjacency that means each node consists of a right away pointer that points to the adjacent node. various records may be traversed by this system.. Graph databases provides schema less and economical storage of semi structured data. The queries are expressed as traversals, therefore creating graph databases quicker than relational databases. it's simple to scale and whiteboard friendly. Graph databases are ACID compliant and provide rollback support. Graph databases may be used for a range of applications like social networking applications, recommendation software system, bioinformatics, content management, security and access management, network and cloud management etc. it's terribly troublesome to attain ‘sharding’ in Graph databases. Graph databases are troublesome to cluster. Neo4j is one in every of the notable graph data stores.
4.1 Neo4j MongoDB

Neo4j MongoDB was developed by modern Neo Technology and was firstly released in 2007. it absolutely was developed by Java. it's a high performance graph data store that provides object directed, versatile network structure. it's supported a Property graph information model that includes of nodes and relationship together with their properties. it's reliable, ACID compliant, extremely out there and durability. It offers REST interface and Java API quiet convenient to use. It can even be embedded into jar files. It uses CYPHER as its source language. Neo4j should be employed in software system.

Example

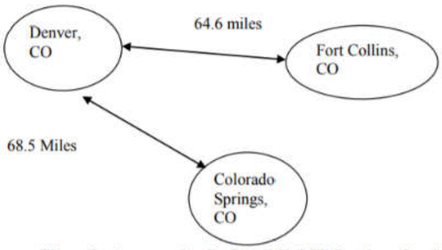


Figure 2. An example of a Graph NoSQL Database for the distance between cities.

Differences between NoSQL data models

Riak is an open-source database developed using Erlang. It has a flexible data schema, offers high availability, tolerance and persistence, but it should be avoided in the case a highly centralized data storage project where the data has a fixed structure.

Cassandra was developed by using Java language and it uses the concepts of key – value pair stores and column–family database stores. It also offers high availability, tolerance, persiste4nce and high scalability. The disadvantage of that it consumes more time to read data than to write data to it. It has its query language CQL which is similar to SQL. MongoDB, a document database, was developed using C++. It has high performance and besides consistency, just like Riak. Cassandra also offers tolerance and persistence. Besides, it provides aggregation, ad hoc queries, indexing etc. BSON format, the format in which the documents are stored, is very efficient both in storage space and scan speed. One disadvantage is that indexing takes up a lot of rams.

Neo4j, developed by Neo Technology using Java, is a high performance graph database. It provides a flexible network structure, it offers high scalability. It uses Cypher query language and queries are presented as a graph traversal. It must be avoided if there are no relationships among the data

Advantages of NoSQL

- Can be used as Primary or Analytic Data Source
- Big Data Capability
- No Single Point of Failure
- Easy Replication
- No Need for Separate Caching Layer
- It provides fast performance and horizontal scalability. Can handle structured, semi-structured, and unstructured data with equal effect
- Object-oriented programming which is easy to use and flexible

- NoSQL databases don't need a dedicated high-performance server
- Support Key Developer Languages and Platforms
- Simple to implement than using RDBMS
- It can serve as the primary data source for online applications.
- Handles big data which manages data velocity, variety, volume, and complexity
- Excels at distributed database and multi-data center operations
- Eliminates the need for a specific caching layer to store data
- Offers a flexible schema design which can easily be altered without downtime or service disruption

Disadvantages of NoSQL

- No standardization rules
- Limited query capabilities
- RDBMS databases and tools are comparatively mature
- It does not offer any traditional database capabilities, like consistency when multiple transactions are performed simultaneously.
- When the volume of data increases it is difficult to maintain unique values as keys become difficult
- Doesn't work as well with relational data
- The learning curve is stiff for new developers
- Open source options so not so popular for enterprises.

5.Conclusion and Future Work

This review paperis written to provide a better overview of the NoSQL databases and data models and to help people learn a bit more about them to data model suits them best. NoSQL does not have the standard query language, and each of these four basic database management systems has developed its own query language, way of structuring and providing data. NoSQL databases not only differ in their provided data model, they also differ in the their offered query functionalities. Databases developed in the last fewyears are very heterogeneous - they differ in their data model, query languages and APIs. Therefore, some research is done in order to achieve an adoption of NoSQL systems by developing standardized query languages for some of thestores. Up to now developers still have to deal with the specific characteristics of each NoSQL store.

Since there is no common query language is available, every NoSQL database differs by its query feature set. Riak executes queries over the key, Cassandra supports its own CQL query language, MongoDB uses JavaScriptlanguage and Neo4j uses Cypher query as its language.

In the future work we will make own tests based on NoSQL databases, which will include comparing performance between different models of NoSQL databases as well as flexibility of their dataschema, availability, consistency and scalability.

“Every job has its tool” is the ideology of the NoSQL community, because every NoSQL database is keenly focused on certain use cases. We can conclude that, NoSQL systems are not ideal because there is a need for a common query language such as SQL, which could be used for all NoSQL database models, and thus make it easier for users to work with them. If such common language appears one day, it is likely that the number of users will increase, and one of the assumptions is that such databases may become more popular than relational databases.

References

- Anić, V. (2016) NoSQLbazapodatakaračunalnihkomponenti, [online], Available at: https://zir.nsk.hr/islandora/object/ etfos:850 [Accessed 4 May 2018]
- Gačić, J. (2017) NoSQLbazepodataka, [online],
- https://zir.nsk.hr/islandora/object/pmf:3234 [Accessed 20 February 2019]
- George, Dr. S. (2013) NoSQL – NotOnly SQL, International Journal of Enterprise Computing and Business Systems, Vol. 2 [online], Available http://www.ijecbs.com/July2013/3.pdf [Accessed 3 May 2018]
- Janković, O. (2015) NoSQLgrafbazapodataka: od domena do modelaprekoupita, Vol. 14, [online], Available at: http://infoteh.etf.unssa.rs.ba/zbornik/2015/radovi/RSS-3/RSS-3-2.pdf [Accessed 3 May 2018]

Author Profile

Neenamol Joseph, Received Degree in Bachelor of Computer Applications from Mahatma Gandhi University Kottayam, Kerala in 2020.

Ponnu Mathew, Received Degree in Bachelor of Computer Applications from Mahatma Gandhi University Kottayam, Kerala in 2020.

Prof. Gibin George, Assistant Professor in Santhigiri College of computer science Vazhithala, India