AN ADVANCED COMPARATIVE STUDY OF THE MOST PROMISING NOSQL AND NEWSQL DATABASES WITH A MULTI-CRITERIA ANALYSIS METHOD

[1] OMAR HAJOUI, [2,1] RACHID DEHBI, [1] MOHAMMED TALEA, [1] ZOUHAIR IBN BATOUTA

[1]Hassan II University, Faculty of Science Ben M'Sik, LTI Laboratory, MOROCCO

[2]Hassan II University, Faculty of Science Aïn Chock, LIAD Laboratory, MOROCCO

E-mail: hajouio@yahoo.fr , dehbirac@yahoo.fr, taleamohamed@yahoo.fr , zouhair.ibnbatouta@gmail.com

ABSTRACT

Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process data within a tolerable elapsed time[1].Big data "size" is a constantly moving target, as of 2012 ranging from a few dozen terabytes to many petabytes of data. Big data is a set of techniques and technologies that require new forms of integration to uncover large hidden values from large datasets that are diverse, complex, and of a massive scale [2]. Currently, there are a hundred solutions to the problem of Big Data that can be classified into three categories: NoSQL databases, NewSQL and Search-based systems. One of the major problems often mentioned is the heterogeneity of the languages and the interfaces they offer to developers and users. Different platforms and languages have been proposed, and applications developed for one system require significant effort to be migrated to another one [3]. Our motivation to write this article is to make a comparative study of Big Data systems, this is our first step to design and implement concrete and effective solution to the interoperability problem between Big Data systems. However, this study will help the professionals in decision-making.

Keywords: *Big Data Analysis, NoSQL Databases, NoSQL Databases Categories, NewSQL Databases, Multi-Criteria Analysis Method (ROC).*

1. INTRODUCTION

With the constant growth of stored and analysed data, classic relational database management systems (RDBMS) exhibit a variety of limitations. Data querying loses efficiency due to the large volumes of data, as well as storage and management of larger databases becomes challenging. NoSQL databases were developed to provide a set of new data management features while overcoming some limitations of currently used relational databases [4]-[5]. NoSQL databases are not relational and they don't require a model or structure for data storage, which facilitates the storage and data search. Also, they allow horizontal scalability, it gives administrators the ability of increasing the number of server machines to minimize overall system load. The new nodes are integrated and operated in an automatic manner by the system. Horizontal scalability reduces the response time of queries with a low cost. It is more beneficial to use such a distributed system instead of building mainframes with high-end capacities. In comparison to relational models, the cost per request or amount of stored data is reduced. Since database administration may be a difficult task with such amounts of data, NoSQL databases are projected to automatically manage and distribute data, recover from faults and repair the whole system automatically [6].

Big Internet actors, especially Google (BigTable), Amazon (Dynamo), LinkedIn (Voldemort), Facebook (Cassandra and HBase), SourceForge.net (MongoDB), Ubuntu One (CouchDB), design and operate NoSQL databases. Other smaller players are making great success (Redis, Riak ,MongoDB ,NuoDB). A significant proportion of these projects is open source and under free license.

NoSQL databases can be used alone or as a complement to a relational database, they increase performance while bringing different advantages and new features. Currently, there are over 150 NoSQL databases with various features and optimizations [4].

NewSQL is a class of modern relational database management systems that seek to provide the same scalable performance of NoSQL systems for online transaction processing(OLTP) read-write workloads while still maintaining the ACID guarantees of a traditional database system [5]. Example systems in this category are NuoDB, VoltDB , Google Spanner , Clustrix .

Since there are different types of NoSQL databases, in order to choose a database that would be more appropriate for a specific business, it is important to understand its main characteristics. Similarly to relational databases, each NoSQL database provides different mechanisms to store and retrieve data, which directly affects performance. Each non-relational database has different optimizations, resulting in different data loading time and execution times for reads or updates.

2. RELATED WORK

During the last four years, several comparative studies of NoSQL databases were made by researchers [7]-[8]-[9]-[10]-[11]. But, these studies do not address the performance which is a crucial criterion in the choice of a Big Data system. Others prefer evaluating just the performance [12]-[13] or on the Cloud [14]. In our study, we decided to consider the main criteria already established by these researchers: performance, integrity, reliability, interoperability, cloud support, query complexity and security. We performed this comparison using the multi-criteria analysis method (ROC).

In this article we compare 10 systems: MongoDB, CouchDB, Cassandra, base, Redis, Riak, Neo4j, Orient DB are NoSQL databases, VoltDB, NuoDB are NewSQL databases. The choice of these systems is based on the classification and the criteria mentioned in the website of Solid IT company [15].

3. CHARACTERISTICS OF NOSQL DATABASES

In computer science,ACID(Atomicity,Consistance , Isolation, Durability) is a set of properties that guarantee that database transactions are processed reliably. In the context of databases, a single logical operation on the data is called a transaction. For example, a transfer of funds from one bank account to another, even involving multiple changes such as debiting one account and crediting another, is a single transaction [16]. Currently all RDBMS implement the ACID properties. However, in a distributed environment, the CAP theorem, also known as Brewer's theorem, states that it is impossible for a distributed computer system to simultaneously provide all three of the following guarantees:

- Consistency (all nodes see the same data at the same time)
- Availability (a guarantee that every request receives a response about whether it succeeded or failed)
- Partition tolerance (the system continues to operate despite arbitrary partitioning due to network failures)

The CAP-Theorem postulates that only two of the three different aspects of scaling out are can be achieved fully at the same time [17].

Many of the NOSQL databases above all have loosened up the requirements on Consistency in order to achieve better Availability and Partitioning. This resulted in systems know as BASE (Basically Available, Soft-state, Eventually consistent). These have no transactions in the classical sense and introduce constraints on the data model to enable better partition schemes (like the Dynamo system) [18].

The general characteristics of NoSQL database are:

- schema-less storage or with dynamically changing schema’s
- complex data structures
- distributed data: horizontal partitioning of data across multiple nodes (servers) usually by the use of MapReduce algorithms
- sharding, replication, fault-tolerance
- data (row) versioning

4. NOSQL DATABASES CATEGORIES

There are several types of NoSQL databases which differ in their data storage models. NOSQL can be broken into 4 different categories: (1) Key-Value stores; (2) Document databases (or stores); (3) Wide-Column (or Columnar) stores); (4) Graph databases .In this chapter, we present the main characteristics, the strengths and weaknesses of each category.

4.1 Key Value Stores

4.1.1 Main characteristics

- data is simply represented by a key / value pair
- both the key and the value can be of any structure
- their model can be likened to a distributed hash table
- is based on four operations (CRUD):
  - **C** reate : creates a new object -> **create(key, value)**
  - **R** ead : reads an object from its key -> **read(key)**
  - **U** pdate : updates the value of an object from its key -> **update(key, value)**
  - **D** elete: removes an object from its key -> **delete(key)**
- simple HTTP Query Interface
- REST : accessible from any programming language
- very high performance in reading and writing
- The most well-known implementations: Amazon Dynamo (Amazon project), Redis (VMware),Riak(Basho),Voldmort(LinkedIn).

Table 1: Key Value Store Example [19]

Key	Value
"John Smith"	"100 Century Dr. Alexandria VA 22304"
"John Doe"	"16 Kozyak Street, Lozenets District, 1408 Sofia Bulgaria"

4.1.2 Applications

- data storage with simple querying needs
- profiles, user preferences
- data from cart of purchases
- sensor data, logs ...

4.1.3 Swot analysis

Table 2: Swot Analysis for Key Value Store [19]

Strengths	**W**eaknesses
Simple Data Model Horizontal Scalability Works well for volatile data High throughput, typically returned, not just a part of it Optimized for reads or writes Simple Protocols to each other	Keys typically are not related To each other The entire value must be returned , not just a part of it Hard to support reporting, analytics, aggregation or ordered values Generally does not support updates in place

4.2 Wide-Column Store

4.2.1 Main characteristics

- stores data tables as sections of columns of data [rather than as rows of data]
- data stored together with meta-data [typically including row id, attribute name & value, timestamp]
- more efficient when an aggregate needs to be computed over many rows but only for a notably smaller subset of all columns of data [because reading that smaller subset of data can be faster than reading all data]
- more efficient when new values of a column are supplied for all rows at once [because that column data can be written efficiently and replace old column data without touching any other columns for the rows]

Table 3: Wide-Column Store Example [20]

Row ID	Columns...		
1	**Name**	**Website**	
	Preston	www.example.com	
2	**Name**	**Website**	
	Julia	www.example.com	
3	**Name**	**Email**	**Website**
	Alice	example@example.com	www.example.com

4.2.2 Applications

- search optimization
- processing of structured data and Business Intelligence (BI)
- high number of fast writes and basic analysis in real time (Cassandra)
- can be good analysis store for semi structured data
- event logging

4.2.3 Swot analysis

Table 4: Swot Analysis for Wide-Column Store [19]

Strengths	Weaknesses
Data model supporting semi-structured data Naturally indexed (columns) Low latency Can be more efficient than row databases when processing a limited number of columns over a large amount of rows	Is much less efficient when processing many columns simultaneously Joins tend to not be supported Referential integrity not available

4.3 Document Store

4.3.1 Main characteristics

- Document stores extend the Key-value paradigm , but the value is a document
- The document tend to semi-structured data (XML,JSON)
- documents can be very heterogeneous in the database
- Complex HTTP Query Interface
- RESTful
- The most well-known implementations : MongoDB,CouchDB (Apache foundation)

Table 5: Document Store Example [19]

Key	Value
"John Smith"	"**<address>** **<street>**100 Century Dr.**</street>** **<city>**Alexandria**</city>** **<state>**VA**</state>** **<postalCode>**22304**</postalCode>** **</address>**"
"John Doe"	"{"**address**":{"**street**": "16 Kozyak Street" "**district**": "Lozenets, 1408" "**city**": "Sofia" "**country**": "Bulgaria" }}"

4.3.2 Applications

- Content Management Systems
- Real-time Web analytics
- Products Catalog

4.3.3 Swot analysis

Table 6: Swot Analysis for Document Store [19]

Strengths	Weaknesses
Tends to support a more complex data model than key/value Good at content management Usually supports multiple indexes Schemaless (can be nested) Typically low latency reads	The entire value must be returned , not just a part of it Joins are not available within the database

4.4 Graph Databases

4.4.1 Main characteristics

- They allow modelling, storage and manipulation of complex data linked by non-trivial or variable relationships
- Data representation model based on graph theory
- Data is stored in terms of nodes and links, both can have attributes.
- The most well-known implementations :**Neo4J , OrientDB** (Apache)

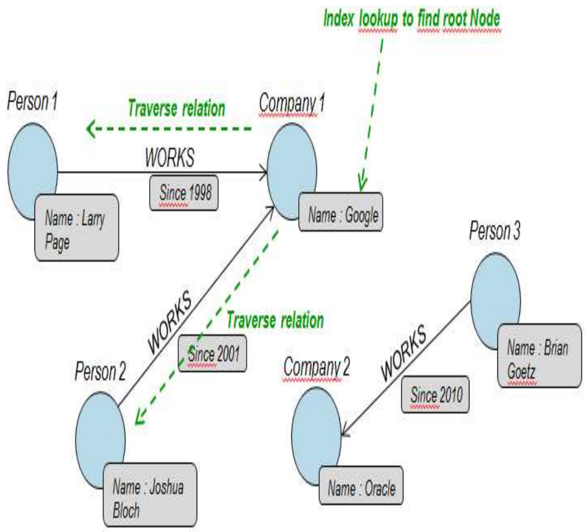

Figure 1: Graph Databases Example [21]

4.4.2 Applications

- Generating recommendations
- Business Intelligence (BI)
- Semantic Web , Social computing
- Geospatial Data , geolocation
- Genealogy, Web of things, Routing Services

4.4.3 Swot analysis

Table 7: Swot Analysis for Graph Databases [19]

Strengths	Weaknesses
They allow deep traversals faster than relational Allow for very fast execution of complex pattern matching queries. Allow for a compact representation of the data, basing its implementation on bitmaps. Most of them are transactional	It is a growing technology that will be mature in a few years, so, you have to bet for one and pray for it becoming successful. Most of them do not have a declarative language You have to use an API Many lack native implementations for different platforms

5. COMPARISON OF DATABASES

In this study, we decided to compare eight NoSQL databases and two NewSQL databases , the choice of these systems is based on the classification given in the dbengines website (solid IT,2015) , we chose the top ranked databases of each approach (Key-Value stores, Document databases ,Wide-Column stores, Graph databases and NewSQL databases).

5.1. Score Computation Method

To calculate the score of a database, we use several criteria, including*: performance, integrity, reliability, interoperability, cloud support, query complexity and security.*

To determine the weight associated with each metric we used a multi-criteria analysis method [22]

5.1.1. Multi-criteria analysis method

Multi-Criteria Decision Analysis, or MCDA, is a valuable tool that can be applied to many complex decisions.

It can solve complex problems that Include qualitative and/or quantitative aspects in a decision-making process.

5.1.2. Why use multi-criteria analysis in database assessment:

The score of a database is calculated based on a number of criteria that the list is not exhaustive. So far we have identified six: *performance, integrity, reliability, interoperability, cloud support, query complexity and security.*

The global score is obtained by adding the partial scores (criteria) affected by relative weights.
In decision analysis, this operation is called synthesis or additive aggregation.

Regarding the assessment of the relative weights of the criteria, there are several Multi-criteria Decision Analysis methods. We selected Rank Order Centroid (ROC) [23] for its simplicity and its proven efficiency.

5.2. Rank Order Centroid (ROC)

Several methods for selecting weights, including equal weights (EW) and rank-order centroid (ROC) weights, have been proposed and evaluated [34].

A common conclusion of these studies is that ROC weights appear to perform better than the other rank-based schemes in terms of choice accuracy.

This method is a simple way of giving weight to a number of items ranked according to their importance. The decision-makers usually can rank items much more easily than give weight to them.

The centroid method assigns weights as follows, where w1 is the weight of the most important objective, w2 the weight of the second most important objective, and so on

$$D = \left\{ W_1 \geq W_2 \geq \ldots \geq W_m \geq 0 \text{ et } \sum_{j=1}^{m} W_j = 1 \right\}$$

This method takes those ranks as inputs and converts them to weights for each of the items.
The conversion is based on the following formula:

$$W_j = \frac{1}{m}\left(\frac{1}{j} + \frac{1}{j+1} + \ldots + \frac{1}{m} \right)$$

5.3. Calculation of Weight by the Classification Rank Order Centroid

Step 1: Sort criteria in descending order of importance: Performance > integrity > reliability > interoperability >cloud support > query complexity > security.

The table 1 of annex A shows how to calculate the weight of each criteria by ROC method. The column control ensures that all weights are normalized (sum of weights = 1).

As result we have the weight of each criteria as shown in the table 2 of annex A.

Therefore, the global score calculation formula is:
Global_Score=(**0,37***performance)+(**0,23***integrity)+(**0,16***reliability)+(**0,11***interoperability)+(**0,07*** cloud_support)+(**0,04*** query_complexity)+(**0,02*** security).

5.4. Database Score Computation

The table 3 of annex A shows a comparison between the different NoSQL and NewSQL databases according to a set of most important features[15]-[24]-[25] -[26]-[27]-[28]-[29]-[30] -[31]-[32]-[33]. Each features group used to evaluate each look database of a given criteria. For example, the data storage and the implementation language features are performance indicators. Which will allow us to assign a score to each database compared to the other for each criteria, we adopt the following scoring system:

- Of equal importance (1)
- Moderately important (3)
- Much more important (5)
- Considerably more important (7)
- Of overwhelming importance (9)

Although this method is not objective, it reflects 90% of reality.

The table 4 of annex A shows the scores for each database for each criteria and the global score calculated using the following formula:

Global_Score=(**0,37***performance)+(**0,23***integrity)+(**0,16***reliability)+(**0,11***interoperability)+(**0,07*** cloud support)+(**0,04*** query complexity)+(**0,02*** security).

We illustrate these results by the following graph:

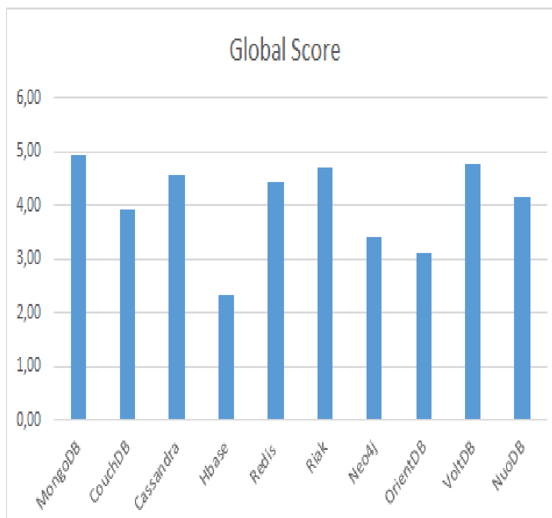

Figure 2: Comparison of Databases Based on Global Score

We note that the MongoGB system has obtained the best score. Cassandra , Redis, Riak and VoltDB have very similar scores of MongoDB. So choosing the right system depends on the most important criteria for the company. The graph in Figure 1 of annex B shows a comparison of ten databases based on each criteria.

6. CONCLUSIONS AND FUTURE WORK

Most NoSQL and NewSQL systems allow storage and efficient management of databases for Big Data, in this papers, we discussed the different approaches of NoSQL databases, the strengths, weaknesses and the main uses of each approach. Also, we compared the eight best NoSQL databases and two promising NewSQL databases on the following criteria: performance, integrity, reliability, interoperability, cloud support, query complexity and security.

We studied the most important technical characteristics of each system, to assess the level of each database in a given criteria. Subsequently, we applied multi-criteria analysis method ROC to calculate the global score of each system. Although Mongodb,Cassandra ,Redis,Riak and VoltDB have almost similar scores, but choosing the right system depends on the customer's need As future work, we will develop a platform of interoperability between databases for Big Data. This platform will enable the exchange and analysis of data stored in multiple Big Data environments. Also, we will investigated the possibility of installing the platform on the cloud.

REFRENCES:

[1] Snijders, C.; Matzat, U.; Reips, U.-D. (2012). "'Big Data: Big gaps of knowledge in the field of Internet". *International Journal of Internet Science* 7: 1–5.

[2] Ibrahim; Targio Hashem, Abaker; Yaqoob, Ibrar; Badrul Anuar, Nor; Mokhtar, Salimah; Gani, Abdullah; Ullah Khan, Samee (2015). "big data" on cloud computing: Review and open research issues". *Information Systems* 47: 98–115. doi:10.1016/j.is.2014.07.006

[3] Paolo Atzeni, Francesca Bugiotti, and Luca Rossi , "Uniform access to non-relational database systems: the SOS platform",(2012), *Advanced Information Systems Engineering.*

[4] nosql-database(2015),*http://nosql-database.org/*

[5] wikipedia(2015) *, http://en.wikipedia.org/wiki/*

[6] Gajendran, S.: A Survey on NoSQL Databases, 2012,*http://ping.sg/story/A-Survey-on-NoSQLDatabases--Department-of-Computer-Science*

[7] Clarence J M Tauro, Aravindh S, Shreeharsha A.B:Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Databases. *International Journal of Computer Applications* (0975 – 888). Volume 48– No.20, June 2012.

[8] Clarence J. M. Tauro1, Baswanth Rao Patil and K. R. Prashanth : A Comparative Analysis of Different NoSQL Databases on Data Model, Query Model and Replication Model. Elsevier Publications. *Proceedings of International Conference on "Emerging Research in Computing ,Information, Communication and Applications",*ERCICA 2013 , ISBN : 9789351071020

[9] A B M Moniruzzaman and Syed Akhter Hossain :NoSQL Database:New Era of Databases for Big data Analytics-Classification, Characteristics and Comparison *.International Journal of Database Theory and Application.* Vol. 6, No. 4. 2013.

[10] Tudorica, B. G., & Bucur, C. (2011, June). A comparison between several NoSQL databases with comments and notes.*In Roedunet International Conference (RoEduNet)*, 2011 10th (pp. 1-5). IEEE.

[11] Han, J., Haihong, E., Le, G., & Du, J. (2011, October).Survey on nosql database. *In Pervasive Computing and Applications (ICPCA), 2011 6th International Conference* on (pp. 363-366). IEEE.

[12] Veronika Abramova,Jorge Bernardino, Pedro Furtado :Which NoSQL Database? A Performance Overview. *Open Journal of Databases (OJDB)* ,Volume 1, Issue 2, 2014

[13] Veronika Abramova,Jorge Bernardino, Pedro Furtado: EXPERIMENTAL EVALUATION OF NOSQL DATABASES.

[14] Dayne Hammes, Hiram medero , Harrison Mitchell: Comparison of NoSQL and SQL Databases in the Cloud , *Proceedings of the Southern Association for Information Systems Conference,* Macon, GA, USA March 21st–22nd, 2014

[15] *http://db-engines.com/en/ranking_definition*

[16] wikipedia(2015) , *http://en.wikipedia.org/wiki/*

[17] Julian Browne on January 11, 2009, Brewer's CAP_Theorem,web*:http://www.julianbrowne.com/article/viewer/brewers-cap-theorem*

[18] infoq (2015), Graph Databases, NOSQL and Neo4j,*http://www.infoq.com/articles/graph-nosql-neo4j*

[19] *http://fr.slideshare.net/cscyphers/big-data-platforms-an-overview*

[20] Nosqlguide(2015),*http://nosqlguide.com/column-store/nosql-databases-explained-wide-column-stores/*

[21] Octo (2015) , *http://blog.octo.com/en/graph-databases-an-overview/*

[22] Roy,B.(2005). An overview of MCDA techniques today: paradigms and challenges. In: Figueira, J., Greco, S. and Ehrgott, M. (eds) Multiple criteria decision analysis: state of the art surveys.

[23] Barron, F.H. 1992. Selecting a best multiattribute alternative with partial information about attribute weights. *Acta Psychologica,* 80: 91-103

[24] Basho(2015) , *http://basho.com/riak/*

[25] cassandra(2015), *cassandra.apache.org/*

[26] couchdb (2015) , *couchdb.apache.org/*

[27] Hbase (2015), *hbase.apache.org/*

[28] mongodb (2015) *, https://www.mongodb.org*

[29] neo4j(2015), *http://neo4j.com/*

[30] Nuodb(2015) , *www.nuodb.com/*

[31] orientdb(2015), *http://orientdb.com/*

[32] Redis(2015), *http://redis.io/*

[33] voltdb(2015), Stonebraker , *http://voltdb.com/*

[34] Butler J, Olson DL. Comparison of centroid and simulation approaches for selection sensitivity analysis. *Journal of Multi-Criteria Decision Analysis* 1999;8:146–61.

Annex A:

Table 1: Calculation of Criteria Weights

	Performance	Integrity	reliability	interoperability	cloud support	query complexity	security	Control
R1	1,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00
R2	0,50	0,50	0,00	0,00	0,00	0,00	0,00	1,00
R3	0,333333333	0,333333	0,333333	0,00	0,00	0,00	0,00	1,00
R4	0,25	0,25	0,25	0,25	0,00	0,00	0,00	1,00
R5	0,2	0,2	0,2	0,2	0,2	0,00	0,00	1,00
R6	0,166666667	0,166667	0,166667	0,166666667	0,16666667	0,166666667	0,00	1,00
R7	0,142857143	0,142857	0,142857	0,142857143	0,14285714	0,142857143	0,142857	1,00
Average	0,37	0,23	0,16	0,11	0,07	0,04	0,02	1,00
								1,00

The column control ensures that all weights are normalized (sum of weights = 1)

Table 2: Criteria Weights

	Performance	Integrity	reliability	interoperability	cloud support	query complexity	security	Control
Weight	0,37	0,23	0,16	0,11	0,07	0,04	0,02	1,00

Table 3: Comparison Some of Databases (five categories) With a Matrix on Basis of Some Technical Attributes.

Category		NoSql Databases								NewSql Databases	
Database model		Document Store		Wide-Column Store		Key-Value Store		Graph-Oriented			
performance	features	MongoDB	CouchDB	Cassandra	Hbase	Redis	Riak	Neo4j	OrientDB	VoltDB	NuoDB
	Data Storage	memory mapped files	append only on disc	disc based data structure , In-memory option on datastax	HDFS	In-memory , append only log	BitCask levelDB In-memory	memory mapped IO	Plocal disc based storage	In-memory ,disc based storage	In-memory ,disc based storage
	Implementation	C++	Erlang	JAVA	JAVA	C	Erlang	JAVA	JAVA	JAVA	C++
interoperability	APIs and other access methods	Custom, binary (BSON)	HTTP/REST	CQL3 & Thrift	JAVA APIs HTTP/REST (also Thrift)	Telnet-like, binary safe	HTTP/REST or custom binary	JAVA APIs HTTP/REST Cypher query language	binary, HTTP REST/JSON, JAVA APIs embedding	JAVA APIs HTTP/REST JSON APIs	HTTP/REST
	Supported programming languages	C,C#,C++,Erlang ,java, php javascript ,python and others	C,C#,C++,Erlang, java,php, javascript ,python and others	C#,C++,Erlang, java,php, javascript ,python and others	C,C#,C++,Erlan g, java,php ,python and others	C,C#,C++,Erlang , java,php javascript ,python and others	C,C#,C++,Erla ng, java,php javascript ,python and others	.NET, java,php javascript ,python and others	.NET,C,C#,C++ , java,php javascript ,python and others	C#,C++,Erlang, java,php javascript ,python and others	.NET,C,C++, java,php javascript ,python and others
cloud	Database as a Service (DBaaS)	yes	yes	yes	no	yes	yes	yes	yes	yes	yes
Query Complexity	SQL	no	no	no	no	no	no	no	SQL Like	yes	yes
	Triggers	no	yes	yes	yes	no	yes	yes	hooks	no	yes
	MapReduce	yes	yes	yes	yes	no	yes	no	no	no	no
	Secondary indexes	yes	yes	only equality queries	no	no	with restrictions	yes	yes	yes	yes
reliability	Partitioning methods	Sharding	Sharding with a proxy based framework	Sharding with no single point failure	Sharding	none	Sharding with no single point failure	none	Sharding	Sharding	data is dynamically stored/cached on the nodes
	Replication methods	Master-slave replication	Master-master replication Master-slave replication	selectable replication factor	selectable replication factor	Master-slave replication	selectable replication factor	Master-slave replication	Master-slave replication	Master-slave replication	yes
integrity	Consistency concepts	Eventual Consistency Immediate Consistency	Eventual Consistency	Eventual Consistency Immediate Consistency	Immediate Consistency	-	Eventual Consistency Immediate Consistency	Eventual Consistency	-	-	Immediate Consistency
	Transaction concepts	atomic operation with a single document possible	atomic operation with a single document possible	Atomicity & Isolation for single operation	no	optimistic locking	no	ACID	ACID	ACID	ACID
	Concurrency	yes	optimistic locking strategy	yes	yes	yes (data access is serialized by the server)	yes	yes	yes	yes (data access is serialized by the server)	yes(MVCC)
security	User concepts	Access rights for users and roles	Access rights for users can be defined per database	Access rights for users can be defined per object	Access Control Lists (ACL)	very simple password-based access control	no	no	Access rights for users and roles	Users and roles with access to stored	Standard SQL roles/ privileges, Administrative

Table 4: Global Score Computation for the Ten Databases

quality indicators	MongoDB	CouchDB	Cassandra	Hbase	Redis	Riak	Neo4j	OrientDB	VoltDB	NuoDB
performance	5	3	5	3	7	5	3	1	5	3
integrity	5	5	5	3	3	5	5	5	5	5
reliability	5	5	5	1	1	5	3	5	5	5
interoperability	5	3	3	1	5	3	3	3	3	3
cloud support	5	5	3	1	3	5	3	3	5	7
Query Complexity	5	3	3	3	5	5	3	5	5	5
security	3	3	5	3	1	1	1	3	5	5
Global Score	4,96	3,92	4,56	2,32	4,42	4,70	3,42	3,12	4,78	4,18

Annex B:

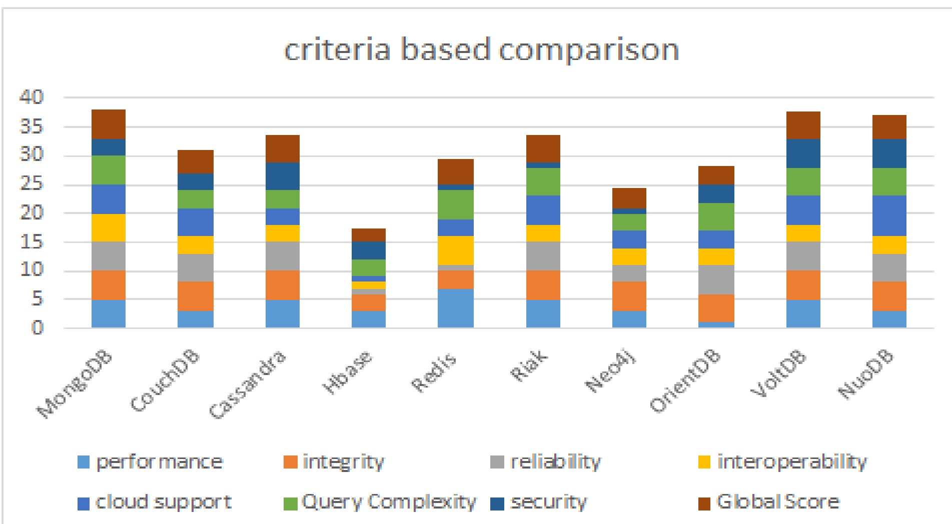

Figure 1: Comparison of Databases Based on Each Criteria