Indonesian Journal of Electrical Engineering and Computer Science
Vol. 3, No. 3, September 2016, pp. 601 ~ 606
DOI: 10.11591/ijeecs.v3.i2.pp601-606

Comparative Analysis of NOSQL Databases

D. Pratiba, Deepak, Shwetha
R V College of Engineering, Visvesvaraya Technological University
Corresponding author, e-mail: pratibad@rvce.edu.in

Abstract

The data being generated is increasing rapidly from day to day, most of it being unstructured data. There is a strong need to store, collect, process and analyze this data. NoSQL databases help us to efficiently deal with unstructured data. The paper deals with classifying the databases using CAP theorem. Further, the paper gives a comparative analysis of NoSQL databases with regard to the applications for which it can be used and how secured each of the databases are.

1. Introduction

The expression unstructured information alludes to the data that doesn't dwell in the customary line section database. Regarding specialized definition, an unstructured information alludes to any data that does not have a pre-characterized information display or is sorted out in a pre-characterized way. Unstructured data often includes text as well as multimedia content. Some of the examples are for text includes e-mail, word documents and some of the multimedia examples include audio, video and image. Even though these content maybe organized internally. They are still considered to be unstructured as they cannot fit into a traditional database. The measure of unstructured information is becoming altogether and is path past organized information's development rate. The current estimate is that 80-90 percent of the data in an organization is said to be unstructured [1].

Strategies, for example, information mining, regular dialect handling and content investigation are utilized to decipher the data from the unstructured information. The Unstructured Information Management Architecture gives a typical system to preparing this data and makes organized information about the data.

The traditional databases store data in terms of tabular relation. These databases are not entirely suitable for storing unstructured data. NoSQL databases are those which give stockpiling and recovery of information in means other than the conventional methodology. The information structures utilized NoSQL databases are more adaptable than the conventional one [2].

1.1. General Classification of NOSQL Databases

There are a lot of NoSQL databases available in the market today and most of them belong to one of the groups mentioned below [3].

1.1.1. Key-Esteem Stores

The key-esteem store use cooperative exhibit as the standard information structure. Here the information is spoken to as a gathering of key-quality sets wherein every key seems just once in the accumulation.

The key-esteem model is one of the least difficult non-unimportant information models, and wealthier information models are frequently executed as an expansion of it. This offers extensive adaptability and all the more nearly takes after cutting edge ideas like item situated programming. Since discretionary qualities are not spoke to by placeholders as in many RDBs, key-esteem stores frequently use far less memory to store the same database, which can prompt huge execution picks up in specific workloads.

Some of the examples are Apache Cassandra, Dynamo.

Received April 2, 2016; Revised July 25, 2016; Accepted August 10, 2016

1.1.2. Document–Oriented Database

The focal idea of an archive store is the thought of a "record. All in all, they all accept that reports epitomize and encode information (or data) in some standard arrangements or encodings. Encodings being used incorporate XML, YAML, and JSON and in addition parallel structures like BSON. Archives are tended to in the database by means of a one of a kind key that speaks to that report. One of the other characterizing attributes of a report arranged database is that notwithstanding the key lookup performed by a key-esteem store, the database offers an API or question dialect that recovers records taking into account their substance.

The report situated databases are intrinsically key-esteem databases. The distinction lies in the way the information is prepared; in a key-esteem store the information is thought to be innately misty to the database, while an archive situated framework depends on interior structure in the report request to concentrate metadata.

Some of the examples are: MongoDB, OrientDB.

1.1.3. Graph Database

This sort of database is intended for information whose relations are very much spoken to as a chart comprising of components interconnected with a limited number of relations between them. The sort of information could be social relations, open transport joins, guides or system topologies. Connections permit the qualities in the store to be identified with each other in a free shape route, rather than customary social databases where the connections are characterized inside the information itself.

These connections permit complex pecking orders to be immediately navigated, tending to one of the more basic execution issues found in conventional key-esteem stores. Most diagram databases likewise include the idea of labels or properties, which are basically connections without a pointer to another record.

Some of the examples are: Neo4j, Ontotext GrpahDB.

1.1.4. Column-Oriented Store

A section of a disseminated information store is a NoSQL object of the most reduced level in a key space. It is a tuple (a key-esteem pair) comprising of three components.

a) Name: Used to identify the column.
b) Value: The content of the column. Include types like ASCII, Long type.
c) Timestamp: The system timestamp to determine the valid content.

Some of the examples are Riak.

2. CAP Theorem

The CAP hypothesis, additionally named Brewer's hypothesis after PC researcher Eric Brewer, expresses that it is unthinkable for a circulated PC framework to all the while give each of the three of the accompanying assurances [4]:

a) Consistency: All hubs see the same information at a given time.
b) Availability: An assurance that each solicitation gets a reaction.
c) Partition Tolerance: The framework keeps on working notwithstanding self-assertive distributing because of system disappointments.
d) The databases are arranged in view of CAP hypothesis as takes after [5, 6]:
e) Concerned about Consistency and Availability: Part of the database is not worried about the allotment resilience, and primarily utilization of Replication way to deal with guarantee information consistency and accessibility. Case: Relational Data Bases.
f) Concerned about Availability and Partition Tolerance: Such frameworks guarantee accessibility and allotment resilience fundamentally by accomplishing consistency, AP's framework. Case: CouchDB.

2.1. The Consistent and Partition Tolerant Data Storage

These systems have master-slave architecture which compromises on availability since all slaves will be useless without master node directing them.

2.1.1. MongoDB

MongoDB [7] is an archive store database which stores semi-organized information written in JSON or JSON like language into BSON (Binary JSON) format.

1. **Features:**
a) It delivers high performance by providing high indexing
b) It scales out without disrupting applications
c) It uses GridFS file system for automatic sharding and storing larger files specially the unstructured multimedia data
2. **Best suited for:**
a) As an alternative to web applications using RDBMS.
b) For providing high scalability and caching operations.
c) For content management of semi-structured data.
3. **Not suited for:**
a) Applications requiring excessive JOIN operations and foreign key referencing.
b) Applications requiring high conformity to ACID properties.

2.1.2. HBase

HBase [8] is Apache's open-source half breed segment situated database which is an impersonation of Google's BigTable.

1. **Features:**
a) It supports auto-sharding and replication data across nodes.
b) It provides automatic load balancing and fault tolerance.
c) It makes use of Bloom Filters (used to test whether a component is an individual from a set)for real time querying
2. **Best suited for:**
a) Applications involving a large number of random reads/write to BigData.
b) Suitable for performing range-based queries.
c) Applications requiring consistency in operations.
3. **Not suited for:**
a) It is an alternative and not a replacement of RDBMS.
b) Applications requiring scanning of extremely large datasets.

2.2. The Available and Partition Tolerant Data Storage

These databases follow a peer to peer architecture unlike the classical master-slave architecture. Therefore they offer high availability at the cost of consistency [9, 10].

2.2.1. Cassandra

Cassandra is a fully distributive open-source column oriented database developed by Apache. It follows a ring architecture where all nodes are independent equal. It ensures partition tolerance and availability offering no single point of failure.

1. **Features:**
a) It offers linear scalability no matter how big the workload is.
b) It can trades-off between consistency with an increase in write latency
c) Reading, writing, updating is extremely simple in
d) Cassandra using built in queries.
2. **Best suited for:**
a) An application where number of reads are more than the number of writes.
b) Web applications that have to provide dynamic schema and content to users, e.g. Netflix.
c) Applications where immediate consistency is not a major concern.
3. **Not suited for:**
a) For transactional and relational operations where high consistency is required.
b) For dynamic querying involving JOIN and Aggregate operations.

2.2.2. CouchDB

CouchDB is an archive situated database written in Erlang and created by Apache. It focuses on the ease of use and completely embraces the web applications [11]. The data is stored in JSON JavaScript with MapReduce to perform queries.

1. **Features:**
a) CouchDB implements multi-version concurrency control which negates the need to lock data during write/update operations.
b) It achieves document level ACID constraints with eventual consistency.
c) For dynamic querying involving JOIN and Aggregate operations.
2. **Best suited for:**
a) It can be an alternative to the MySQL relational database in web applications
b) Applications involving periodic logging such as financing organizations where data is regularly modified and updated on the server.
3. **Not suited for:**
a) Documents are stored in an isolated fashion and operations involving merging or aggregating two documents can be performed on the application layer and not on the database. This can be unfeasible for web developers at times.

3. Securit of NoSQL Databases

Aside from versatility and execution, information security is likely a standout amongst the most troublesome difficulties confronted by the NoSQL databases now-a-days. These databases were at first not planned by considering security as a vital component. In this manner, it has turned into the sole obligation of NoSQL shoppers to secure these databases by utilizing outsider apparatuses and services [12].

Disseminating information to numerous servers in various server farms gives more boulevards to both physical and virtual security assaults in this way, it is imperative to distinguish the elements essential for implementing security in sharded databases and to consider that these elements ought to be material to each database similarly. A brief portrayal of every component and the near investigation taking into account every element is given underneath [13]:

3.1. Validation

The way toward confirming a client's personality who needs to get to the assets, information or utilizations of an association is known as validation. Validation can be given from numerous points of view, running from a solitary client verification to shared confirmation of client with database server and after that to two-route verification between database servers.

Figure 1 shows that databases like MongoDB and CouchDB provides high security in terms of authentication.

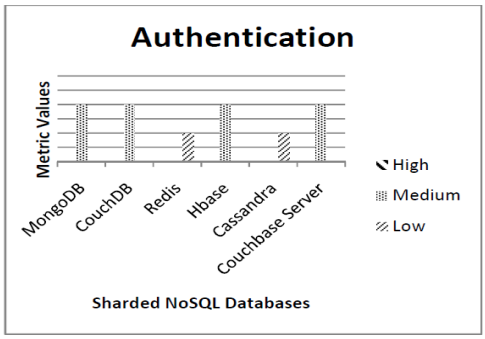

Figure 1. Authentication in Databases

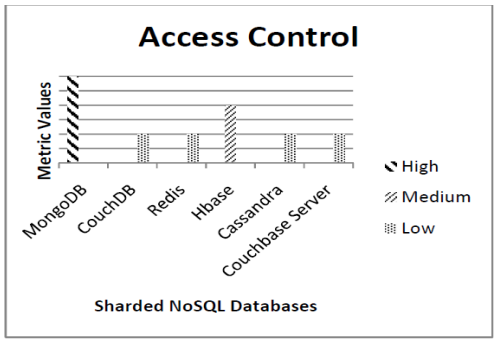

Figure 2. Access Control in Databases

3.2. Access Control

Access control is the instrument through which we can guarantee that exclusive an approved individual is permitted to get to framework assets. Access control can be connected at framework, database, question and substance level contingent upon the arrangements of the database head. Various access control models have been proposed to give secure access to database tables and sections having delicate qualities append to them.

Figure 2 shows how efficient access control is each of the databases. MongoDB seems to be the only database which provides very high access control as a security feature.

3.3. Secure Configurations

A security break may effectively happen because of misconfigurations at the OS, database or application layer along these lines, sharded database must incorporate a rundown of designs that can be connected by the framework heads to secure databases consistently crosswise over groups at both physical and coherent level. These setups are executed in view of the security and business needs of an association.

Figure 3 shows how MongoDB outperforms other databases in terms of secure configuration. Other database seems to lack the feature of secure configurations.

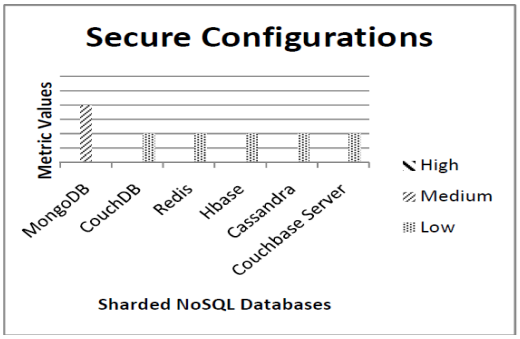

Figure 3. Secure Configurations in Databases

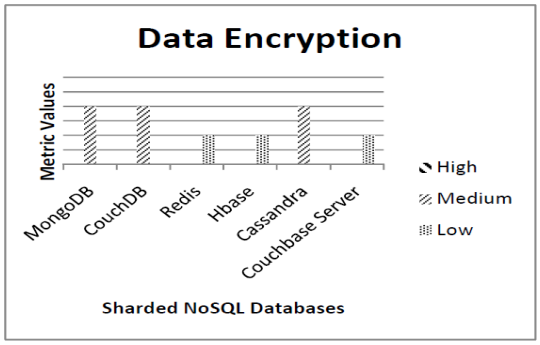

Figure 4. Data Encryption in Databases

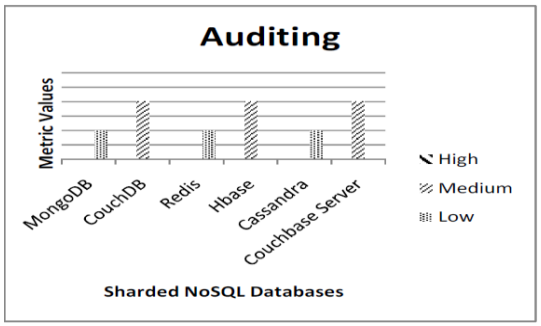

Figure 5. Auditing in Databases

3.4. Data Encryption

Information Encryption is utilized to give secrecy of information and applications in a database framework. It incorporates the encryption of information very still and also the encryption of information in-travel over the system. Sharded databases are expected to apply information encryption procedures at table, line and section level to secure the data. The basic techniques for information very still encryption are the usage of calculations, for example, DES (Data Encryption Standard), AES (Advance Encryption Standard) and Hashing (MD5, SHA1 &2) and so forth.

Figure 4 shows how the data encryption is applied in various databases. We see that most of the database has medium level of encryption and none of them have high encryption standards.

3.5. Auditing

Database Auditing alludes to the checking and recording of individual and aggregate activities performed by database clients. Examining helps in the recognizable proof of impressions or conceivable secret word splitting endeavors before the event of an assault.

Figure 5 below shows how the auditing is done in each database and shows how CouchDB and Hbase have high auditing while others have medium level of auditing.

4. Conclusion

The presence of unstructured data is increasing manifolds when compared to the unstructured data in any organization. The traditional database was not efficient enough to store unstructured data. In this way there was a requirement for on a level plane adaptable, disseminated non-social NoSQL databases.

The paper speculated about the unstructured data and NoSQL databases. The paper focused on the classification of NoSQL databases. The core element is the comparison of NoSQL databases and finding applications for which each NoSQL database is suitable for. Using the CAP theorem we found the advantages and disadvantages of some of the NoSQL databases. MongoDB proved to be one of the most efficient, highly available databases with features that could be used for a variety of modern applications.

In terms of security, MongoDB is said to be more secured than any other databases. The auditing is the only factor where the auditing is medium in MongoDB. The unstructured data can be efficiently handled using NoSQL databases where the storage system is efficient and fast.

References

[1] Moniruzzaman ABM, Syed Akhter Hossain. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. 2013.
[2] Strauch Christof, Ultra-Large Scale Sites, and Walter Kriha. NoSQL databases. Lecture Notes. Stuttgart Media University. 2011.
[3] Z Chen, S Yang, H Zhao, H Yin. *An Objective Function for Dividing Class Family in NoSQL Database*. Proceedings of International Conference on Computer Science & Service System (CSSS). Nanjing. 2012: 2091-2094.
[4] S Gilbert, N Lynch. Perspectives on the CAP Theorem. *Computer*. 2012; 45 (2): 30-36.
[5] PP Srivastava, S Goyal, A Kumar. *Analysis of various NoSql database*. Proceedings of International Conference on Green Computing and Internet of Things (ICGCIoT*).* Noida. 2015: 539-544.
[6] Jing Han, Haihong E, Guan Le, Jian Du. *Survey on NoSQL database*. Pervasive Computing and Applications (ICPCA), 2011 6th International Conference*.* Port Elizabeth. 2011: 363-366.
[7] Chodorow Kristina. MongoDB: the definitive guide. O'Reilly Media Inc. 2013.
[8] George Lars. HBase: the definitive guide. O'Reilly Media Inc. 2011.
[9] Lakshman Avinash, Prashant Malik. *Cassandra: structured storage system on a p2p network*. Proceedings of the 28th ACM symposium on Principles of distributed computing. ACM. 2009.
[10] Lakshman Avinash, Prashant Malik. *Cassandra: a decentralized structured storage system*. ACM SIGOPS Operating Systems Review. 2010; 44(2): 35-40.
[11] Anderson, J Chris, Jan Lehnardt, Noah Slater. CouchDB: the definitive guide. O'Reilly Media Inc. 2010.
[12] L Okman, N Gal-Oz, Y Gonen, E Gudes, J Abramov. *Security Issues in NoSQL Databases*. Proceedings of 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications. Changsha. 2011: 541-547.
[13] Zahid, R Masood, MA Shibli. *Security of sharded NoSQL databases: A comparative analysis*. Proceedings of Conference on Information Assurance and Cyber Security (CIACS*)*. Rawalpindi. 2014: 1-8.