A Brief Study of Comparison between Three Document Databases

Finna Suroso
Information System ManagementGunadarma University

Galih Hendro Martono
Department of InformaticsSTMIK Bumigora Mataram

ABSTRACT

The development of data and volume has increased with the presence of the internet that is able to process and store data in the form of text, images, and videos. The emergence of big data provides a solution for companies to analyze data in real time. One of the most important points in big data is handling large data and volumes with the database. Conventional database concepts with Relational Database Management System (RDBM) models are unable to deal with these problems because they are less flexible in varying data handling.

No SQL is a database used to solve problems from Big Data. There are four types of No SQL, namely Key-Value database, Document Database, Column Family Database, and Graph Database. The difference between them is data handling and processing methods. The document database is the most widely used No SQL database because it's flexibility, easy to use, and similarities with RDBMS. This paper conducts a literature review of document databases, namely Monggo DB, Couch DB, and Couch Base. These three databases are selected because the three of them are the most widely used database. This paper not only compares the three databases in general but also, based on CAP Theorem. The purpose of this paper is to provide an overview of the three databases. Hopefully, this paper not only can give an overview of a document database but also understanding of advantages and disadvantages of each database so in practice users can choose the most suitable database for their need.

Keywords

Document database, Monggo DB, CouchDB, CouchBase, No SQL

1. INTRODUCTION

Big Data is a term used to store large-scale data in the form of structured data or unstructured data. The term Big Data relates to the size of large data storage. Companies like Facebook, Twitter, Google, Amazon, and Yahoo have very large data that requires a database that can flexibly handle large amounts of data storage. The development of the amount of data on the internet today raises a problem as well as a challenge for the company in overcoming very large data settings.

Big Data consists of three important components:

- **Volume**

One of the problems for Big Data is the large volume of data. Problem rise not only because of storage media but also, how to process and analyze the data. Large amounts of data can be obtained from various sources such as social media, the internet, digital images, purchase transaction records, internal company data, government and so on.

- **Speed**

Speed refers to the time needed and the accuracy of the processing of large amounts of data.

- **Variations**

Variations refer to the diversity of the existing data. This happens because of the different platforms used. Differences in source data formats used such as RDF, HTML, XML, and so on so; therefore Big Data needs to support the interoperability in processing the various source of data. Big Data does not only consist of structured data such as numbers or letters but also consists of multimedia data such as images and videos or unstructured data. The data component above is a problem in Big Data, that is how to store data that has large and diverse volume quickly. Large amounts of data storage cannot be processed with conventional database storage such as SQL or RDBMS, for which a storage that supports high availability and scalability such as No SQL is needed.

No SQL was first used by Carlo Strozzi in 1998 [1]. No SQL or Not Only SQL means that the architecture used is different from SQL. No SQL is a new approach to data management and database design for large amounts of distributed data [2]. Structurally, the No SQL Database does not use the relation between tables and does not store data in a rigid table format (a fixed column) like a Relational Database. This makes database storage with No SQL widely used because it supports schema-less so that data can be managed easily and has a simple binary data storage mechanism [3]. There are 4 types of No SQL database models including Key-Value stores, Wide column stores, Graph databases, and Document-Oriented databases or Document stores . This publication will discuss the document-oriented database commonly known as the document database. The document database has a hierarchical data structure where each document has an attribute called key that is used for the unique identification of the document [4]. These documents are encoded in standard data formats such as XML, JSON (Javascript Option Notation) or BSON (Binary JSON) [5]. The document store does not accentuate read and write speeds simultaneously, but rather to ensure that large data storage and query performance is good [6].

2. CHARACTERISTIC OF NO SQL

No SQL which means that the database management system is different from the relational database in several aspects. Some of the advantages of the No SQL database include [7]:

- Horizontally scalable: the No SQL database distributes the load evenly to each host and helps improve performance by increasing data horizontally.
- Schema-free: there is no need to define structures in a data set as in a relational database.
- Low cost: No SQL is open source software.

- Integrated Caching Facility: Store temporary data in system memory to improve its performance and increase data output.

Database related to data providing and processing must consistency and integrated. This is related to the characteristic of the database that is, Atomicity, Consistency, Isolation, and Durability(ACID). ACID make sure every transaction in the database used all or nothing concept to reduce load management when there are many variables included. Support of ACID in both RDMS and No SQL is important because can ensure the safety of data and support parallelism an concurrency.

Atomicity means that the transactions are atomic. That is because a transaction on the database is unified which means that the transaction is carried out in whole or not at all.

Consistency mean database can show the consistency of existing data after a transaction occurs

Isolation means that transactions that are running will not interfere with other transactions. So if there are other operations that will change then you must wait until the transaction is complete.
Durability means the results of the transaction must remain safe and stored. Many DBMS write logs for a transaction that can be used when there is an error in hardware or software.

As the alternative of ACID, the emerging concept of BASE. This concept more flexible than ACID because don't need to fulfill the criteria of the characteristic of the database but close enough to the characteristic. BASE stands for Basically Available, Soft State, Eventually Consistent [9], that is:

- **Basically Available**, this means that if there is a partial failure in some parts of the distributed system, the system can still function to ensure that the system works all the time.

- **Soft state**, In No SQL, there is the fact that data can eventually be overwritten with newer data. This is a system condition not to be consistent at all times.

- **Eventually Consistent**. This means that there may be times when the database is in an inconsistent state. Emphasizing that the system will be consistent sometime later. This can occur if the user or program updates one copy of the data and another copy continues to have an older version of the data. Systems with BASE properties are no longer limited by the CAP Theory, thus offering high horizontal scalability [1].

In 2000, Professor Eric Brewer put forward the CAP Theorem which was famous for the extension of Consistency, Availability, tolerance of network Partition (Tolerance in network partitions) [8].

After the computer scientist put forward the theory, there is a statement that states that distributed databases cannot have all CAPs at the same time [9]. The principle of the CAP theorem is:

- **Consistency**, in this case, means a consistent copy of data on different servers.

- **Availability** refers to responding to any query.

- **Partition** protection means that if a network connecting two or more database servers fails, the server will still be available with consistent data.

It is very difficult to develop error-tolerant BASE software in the world compared to the ACID world, however, as Brewer points out in the CAP Theorem, there is a series between ACID and BASE. Data sharing systems can be carried out by fulfilling two of the three CAP theories [10]. So if referring to the CAP scheme, there are two possibilities of two applications in the CAP theorem, namely:

- **Consistency-Availability (CA)**
The application of this theorem is conducted on a single site cluster so that all nodes are connected so that it does not focus on partition tolerance. The database that is suitable to be applied to this theorem is a database based on relational models (RDBMS)

- **Consistency-Partition Tolerance (CP)**
The application of this theorem is data that has a distributed system, but on the other hand, has consistency and accuracy. The databases included in this theorem are MongoDB, HBase, Memcached, and Redis.

- **Availability-Partition Tolerance (AP)**
The application of this theorem is to allow data partitioning but cause less accuracy in the data. the database that belongs to this theorem is CouchDB, Cassandra, Riak, DynamoDB.

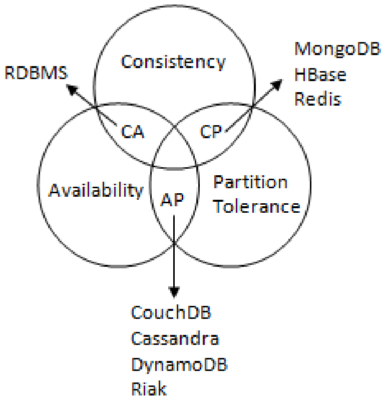

Fig 1: CAP Theorem

3. DOCUMENT STORE

The document database is a No SQL model that is widely used by users when they want to migrate from RDBMS. This is because of the flexible nature of No SQL and does not require a general structure for data storage. Besides that, the document store has features and functions similar to RDBMS. In the document store, each data object is stored in data called documents. The document itself can consist of key-value and it can be an array or multilevel value. Document stores are very useful when creating dynamic databases [11]. From the data obtained from the DB Engine website page [12], it is known that the most widely used document stores are MongoDB, Couchbase, and CouchDB. Comparison of the number of document database users can be seen in Figure 2.

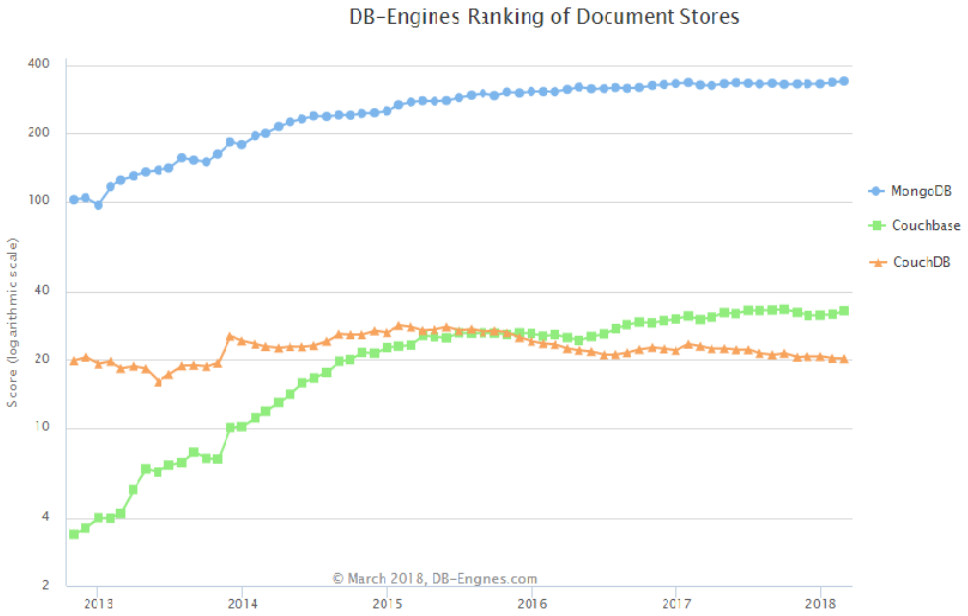

Figure 2. Comparison of Document Store Use

The picture shows that the top three uses of document stores include MongoDB, Couchbase and CouchDB. From 2013 to 2015 the first rank was occupied by MongoDB, both CouchDB and third Couchbase. However, in mid-2015 Couchbase experienced a significant increase. So that the second position is occupied by Couchbase and the third CouchDB while MongoDB still lasts from year to year in the first position in the category document store.

3.1 MongoDB

MongoDB is a document database developed by a company from New York City, 10gen (now MongoDB, Inc.) In October 2007 as part of a platform planned as a service product, the company shifted to an open source model in 2009, with 10gen offering commercial support and other services [13]. Since 2009 MongoDB has been adopted as back-end software by a number of sites and services, including Craigslist, eBay, Foursquare, SourceForge, and The New York Times. MongoDB is the most popular No SQL database on the internet [14]. MongoDB is often used for Cloud-based, Grid Computing, or Big Data applications [15]. The official website is http://www.mongodb.org/.

As one of the No SQL databases, MongoDB uses JSON data structures to store data. The features found in MongoDB are as follows [16] [17]:

- Non-relational database;
- Support complex data types and multiple data structures;
- Formal query languages allow most of the functions such as queries in a single relational database table, and support indexes.
- Quick access to data processing.
- Set-oriented storage and easy to store object types
- Index support
- The sequence of data fragment processing to support cloud level expansion
- Supported by C, C ++, C #, Erlang, Haskell, Java, JavaScript, Perl, PHP, Python, Ruby, and Scala.

MongoDB in terms of usage popularity ranks 5th out of all and No SQL databases rank 1st in the document database category.

3.2 CouchDB

CouchDB is one of the Apache Foundation projects created by Damien Katz in 2008. CouchDB stands for cluster of unreliable commodity hardware. CouchDB has an administrative tool called Futon which has an easy user interface so that it can help developers to operate databases and use JavaScript to query and other database operations [18].

CouchDB is built using the Erlang programming language that relies on reliability and concurrency. The official website of CouchDB is http://couchdb.apache.org. According to S. Zhang [7], CouchDB is a document-oriented database with JavaScript Oriented Notation (JSON) structure. CouchDB has the following features [19]:

- Supports HTTP / JSON API
- Having a Futon GUI that can help manage the database
- Support for incremental replication and conflict management
- Incremental Map with built-in JavaScript support
- Excellent data integrity/reliability
- BLOB Support (Binary Large Objects)
- Easy to install on many platforms, from servers to mobile devices
- Have a user community that can be used to share knowledge
- Supported with C, C #, ColdFusion, Erlang, Haskell, Java, JavaScript, Lisp, Lua, Objective-C, OCaml, Perl, PHP, PL / SQL, Python, Ruby and

Smalltalk CouchDB in terms of usage popularity ranks 28th of all databases and ranks 4th in the document store category [12].

3.3 Couchbase

Couchbase is part of Couchbase Inc. which consists of private companies such as Accel Partners, Adams Street Partners, Ignition Partners, Mayfield Fund, North Bridge Venture Partners, Sorenson Capital, and WestSummit Capital. Couchbase official website is https://www.couchbase.com and Couchbase developer official website is https://developer.couchbase.com/server.

Couchbase is a No SQL database that can support JSON or binary data [16]. Couchbase Server is an open source database that distributes No SQL document databases that provide low latency data management for large-scale interactive online applications. It is designed to scale, especially horizontally easily and without performance degradation. Built with a strong emphasis on reliability, high availability, and simple management, Couchbase presents endless data with minimal human intervention [20]. Couchbase is a combination of two No SQL open source databases, that is CouchDB (document store) and Membase (database key-value) [17]. Couchbase Server specifically provides low latency data management for large-scale interactive, mobile and IoT web applications with the following Couchbase Server features, that is:

- Flexible data model
- Strong query language
- Scalability
- Performance
- Simple administration
- High Availability
- Supported by C SDK 2.4 / 2.5, Go SDK 1.0, Java SDK 2.2, .NET SDK 2.2, Node.js SDK 2.0 / 2.1, PHP SDK 2.0 / 2.1, Python SDK 2.0, and Ruby

Couchbase in terms of usage popularity ranks 23rd of all databases and ranks 3rd in the document store category [12].

4. COMPARISON OF DOCUMENT DATABASE

In this section, a comparison of the three documents database will be reviewed from some literature. Comparison of the document database can be seen from the characteristics of the character or general specific database, CAP theorem, and performance. Document databases are used to handle semi-structured data using slave, asynchronous, and reduce maps for replication. MongoDB stores data with a dynamic schema called BSON which is possible to store data with the same structure so that it is more flexible, in contrast to Couch DB and Couch Base storing data in JSON format. Specific general descriptions of the document database can be seen in table 1.

Table 1. Common Comparison of Document Database

Study	Document Database	Development Language	Data Management	Data Storage	Storage type	Query Method	Replication	Database Applicability	License
[1],[9],[12]	Monggo DB	C ++, C, JavaScript	semi-structured	Disk	BSON (Binary JavaScript Object Notation)/Document	MapReduce / Support MapReduce on sharded collections, to be written in Java Script	Master-Slave / Asynchronous	Forsquare, shutterfly	GNU AGPL v3.0, Mongo DB
[21], [19]	CouchDb	Erlang	semi-structured	Disk	JSON (JavaScript Object Notation)/Document	MapReduce / Supports MapReduce for queries – supports HTTP and REST API	Master-Master (Multi Master) / Asynchronous	BBC	Apache license 2.0
[20],[17]	CouchBase	C++, Erlang, C, Go	semi-structured	Disk	JSON (JavaScript Object Notation)/Document	MapReduce, N1QL, Memcached protocol RESTful HTTP API (only for server administration)	Master-master (including cross data center replication), Master-slave replication	Orbitz	Apache License, freemium

In the database system, it is very difficult to meet the 3 criteria of CAP theorem so that in fact the database is said to be good nugh if it meets 2 of the 3 criteria of the CAP theorem. When viewed from the CAP theorem, for a database document that meets the CAP theorem can be seen in table 2.

Table 2. Document Database based on CAP Feature

Study	Document Database	CAP Feature
	MonggoDB	Consistence and Partition Tolerance (CP)
	CouchDb	Availability and Partition Tolerance (AP)
	CouchBase	Normally a CP type system meaning it provides consistency and partition tolerance, or it can be set up as an AP system with multiple clusters

Compared document database can be seen in several aspects such as system performance, scalability, availability, and data operation. Monggo DB offers lower throughput, automatic scaling, and high availability. Data operation in MongoDB uses javascript to perform operations such as CRUD, aggregation, indexing, other data. CouchDB has almost the same performance as MongoDB but in CouchDB, it doesn't have auto shading facilities. The comparison of the performance of the database can be seen in table 3.

Tabel 3. Document Database Comparison Based on Performance

Study	Document Database	System Performance	Scalability	Availability	Data Operation
[1], [22]	MonggoDB	Lower Throughput	Complex multi-step scalling, nowrite scaling across data centers	High & inconsistence latency	read and write operations (CRUD)
[7],[21]	CouchDB	Higher throughput, and more space efficiency	Incremental	eventual consistency	key-access & disk-only writes
[20],[9]	CouchBase	Consistent High Throughput	With 1-click, horizontally grow cluster, even scale across data centers	Consistent sub-millisecond read/writes	Reading and writing data

5. CONCLUSION

In the digital era today it presents its own challenges in the resolution of big data where the need for fast data processing in data that has a large volume and has a variety of formats. One of the support for Big Data is the use of the No SQL database. This paper describes some of the No SQL database store document that is widely used based on the DB-Engines Ranking of Document Stores namely MongoDB, CouchDB, and CouchBase. In this paper, a literature review is conducted from various sources to compare the three databases in general and seen from the CAP Theorem concept. The MongoDB database is the most widely used document store database. When viewed from the CAP Theorem, Couchbase approaches the concept of CAP theorem when compared to the other two databases because of its consistency in couchbase, partition tolerance and its ability to be run on multiple clusters. Judging from its performance, the CouchDB and couchbase databases have better performance than MongoDB because they are more consistent. However, the document database still has limitations and weaknesses in the future work can develop research about an empirical evaluation focused on to make a qualitative and quantitative analysis for storing data.

6. REFERENCES

[1] CloudAnt, “NO SQL Databases,” 2014.

[2] C. T. Yang, J. C. Liu, W. H. Hsu, H. W. Lu, and W. C. C. Chu, “Implementation of data transform method into No SQL database for healthcare data,” Parallel Distrib. Comput. Appl. Technol. PDCAT Proc., pp. 198–205, 2014.

[3] B. Wylie, D. Dunlavy, W. D. Iv, and J. Baumes, “Using No SQL Databases for Streaming Network Analysis,” IEEE Symp. Large Data Anal. Vis., pp. 121–124, 2012.

[4] M. Qi, “Digital forensics and No SQL databases,” 2014 11th Int. Conf. Fuzzy Syst. Knowl. Discov., pp. 734–739, 2014.

[5] A. B. M. Moniruzzaman and S. A. Hossain, “No SQL database: New era of databases for Big Data analytics-classification, characteristics and comparison,” arXiv Prepr. arXiv1307.0191, vol. 6, no. 4, pp. 1–14, 2013.

[6] C. He, “Survey on No SQL Database Technology,” vol. 2, no. 2, pp. 50–54, 2015.

[7] R. Rani, “CouchDB Document Oriented Databases.”

[8] A. Guidi, H. Gharsellaoui, and S. Ben Ahmed, “A No SQL-based Approach for Real-Time Managing of Embedded Data Bases,” Proc. - 2016 World Symp. Comput. Appl. Res. WSCAR 2016, pp. 110–115, 2016.

[9] D. Sullivan, No SQL for Mere Mortals. 2015.

[10] John D. Cook, “John D. Cook Consulting.” [Online]. Available: https://www.johndcook.com/blog/2009/07/06/brewer-cap-theorem-base/. [Accessed: 16-Mar-2018].

[11] K. M. Hurwitz J., Nugent A., Halper F., Big Data for Dummies. John Wiley and Sons Inc, 2013.

[12] solid IT, “DB-Engines Ranking.” [Online]. Available: https://db-engines.com/de/ranking. [Accessed: 16-Mar-2018].

[13] Gigaom, “10gen embraces what it created, becomes MongoDB Inc,” 2018. [Online]. Available: https://gigaom.com. [Accessed: 16-Mar-2018].

[14] P. P. Srivastava, S. Goyal, and A. Kumar, “Analysis of various No SQL database,” Proc. 2015 Int. Conf. Green Comput. Internet Things, ICGCIoT 2015, pp. 539–544, 2016.

[15] “MongoDB.”[Online].Available: http://www.mongodb.org/display/DOCS/Home.

[16] R. M. Chopade and A. Basics, “MongoDB, CouchBase : Performance Comparison for Image Dataset,” pp. 255–258, 2017.

[17] Y. Fan, “Performance Comparison between Five No SQL Databases,” pp. 117–121, 2016.

[18] D. Katz, “Notes on Building Noise: a JSON Search Engine written in Rust.” [Online]. Available: http://damienkatz.com/. [Accessed: 16-Mar-2018].

[19] Apache Foundation, “Apache CouchDB.” [Online]. Available: https://wiki.apache.org/couchdb/. [Accessed: 16-Mar-2018].

[20] couchbase, “Introduction.” [Online]. Available: https://developer.couchbase.com/documentation/server/4.0/introduction/. [Accessed: 16-Mar-2018].

[21] K. B. Sundhara Kumar, Srividya, and S. Mohanavalli, “A performance comparison of document oriented No SQL databases,” Int. Conf. Comput. Commun. Signal Process. Spec. Focus IoT, ICCCSP 2017, 2017.

[22] I. MongoDB, “Documentation.” [Online]. Available: https://docs.mongodb.com/manual/introduction/. [Accessed: 16-Mar-2018].