International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 4, April - 2014

Analysis and Comparison of NOSQL's Key-Value & Document-Oriented Databases

Tushar Seth
Amity Institute of Information Technology
Amity University
Noida, India

Bhawna Minocha (Guide)
Amity Institute of Information Technology
Amity University Noida, India

***Abstract*— From many years, Relational Databases are being used for organizing data in form of tables or relations. With the increasing amount of data in 4 V's i.e. Volume, Variety, Velocity, Veracity, a new database being introduced named NOSQL (Not Only SQL). Recently NOSQL databases are widely used in many big companies for storing and managing data. NOSQL databases also overcome few drawbacks of SQL. NOSQL databases are more optimized and flexible in comparison to SQL. In this paper, we address the advantage and disadvantage of two broadly used techniques of NOSQL databases used for storage and also compare on their characteristics of the two techniques, they are –Key Value & Document–oriented databases.**

We compare and discuss two techniques- key-value store & Document-oriented databases on different parameters. Each NOSQL database technique has its own advantage and their use. We also discuss and compare few differences between RDMBS and NoSQL databases.

Keywords- NOSQL, Big data, Key-value databases, Document – oriented databases, RDBMS, CAP theorem, ACID, BASE.

1. INTRODUCTION

The term "NoSQL" was first used by Carlo Strozzi in 1998 to name his fast, portable, lightweight shell based -Open source relational database [1]. Carlo used to pronounce it as noseequel. It was first introduced for Unix Operating System by Carlo and latter was developed by Walter.W.Hobbs references in [1]. NoSQL is often interpreted as 'Not SQL' but it should be interpreted as 'Not Only SQL'. NoSQL is a new database which provides a method for storage and retrieval of data in modelled structure like key-value store, document-oriented store, graph, tree etc. It does not use the very old, robust and widely used model Relational Databases. NoSQL becomes popular because of its ability to store large amount of data (Big data), horizontal scaling and simplicity in design structure. It is one of the growing fields in big data and being widely used in distributed and real-time web applications [6].

Many Big Companies are dependent on NoSQL databases like GOOGLE uses BIGTABLE; Amazon uses Amazon Dynamo [2], Memcached at Facebook, Zynga and Twitter[21] [3], Foursquare uses MongoDB, Mozilla and Adobe uses HBASE, Linkedin uses Voldemort etc [5]. NoSQL databases have scalable architecture so it can efficiently scale up to many machines easily with minimum effort. NoSQL is schema-less i.e. it supports Dynamic Schemas .Suppose in future you want to change the length or datatype of column, or add new column then we don't need to change whole table structure instead we are able to store with the new structure without affecting the previous data structure. NoSQL also supports feature like auto-sharding which is one of the most important and useful feature. SQL database generally use to scale its data vertically i.e. single server is used to store database and serve to its application while NoSQL uses horizontal scaling i.e. adding more servers to store data other than storing in single server. Auto-sharding feature enables data to distribute its data in different servers and NoSQL database can automatically manage its data [7]. Adding more servers is an advantage is to increase the capacity and the performance of write and read operations [13]. It is even possible to reduce the size of a sharded database cluster when the demands decrease. Automatic Replication of data i.e. sharing of data between nodes or system are also be supported by many NoSQL databases. So it provides highly availability of data as well as disaster recovery management [11]. This is very useful to increase read performance of the database, because it allows a load balancer to distribute all read operations over many machines. It is also very advantageous if one machine fails, then there is at least another machine with the same data which can replace the lost node.

In this research paper we are going to discuss briefly about the differences between RDBMS with NoSQL and the two mostly used NoSQL techniques i.e. Key-value Store & Document-oriented Store.

Google Trends Showing NoSQL Interest Over a Period of Time [8]

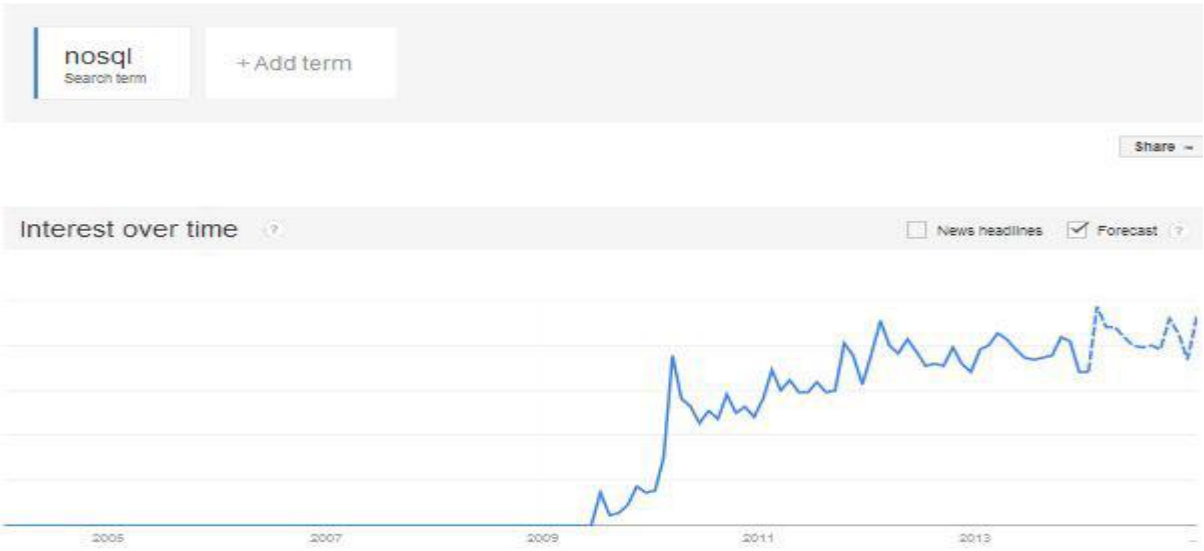

Fig-1: Google Trend Showing Interest over time

2. RDBMS VS NOSQL

Due to the growing field of big data and analytics has introduced a new database NoSQL which is quite useful in distributed and cloud based web applications and beats RDBMS in few areas like fixed schema, JOIN operations , scalability , fast read operation , performance etc[19]. Apart from these advantages NoSQL lacks in providing ACID (Atomicity, Consistency, Isolation and Durability) type transaction support. Berkeley's computer scientists suggested a theorem for distributed environment which is known as CAP theorem or
Brewer's theorem. He said we needed 3 requirements i.e. Consistency, Availability, and Partition Tolerance but we can only fulfil two of the three requirements [18]. Mostly NoSQL databases face consistency problem to achieve scalability. As in cloud –based applications mostly read operation is needed to perform. Instead of satisfying ACID properties it uses BASE approach (Basically Available, Soft-state and eventually consistent) [16][17]. So it means choose any two requirements and do something to manage the third to satisfy all three [14][15].

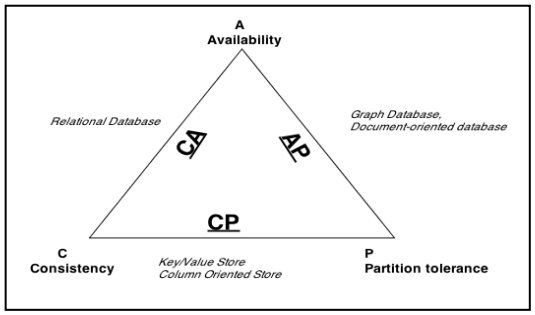

Fig 2: CAP Theorem [19]

ACID	BASE
• Strong consistency • Isolation • Focus on "commit" • Nested transaction • Availability • Pessimistic • Evolution difficult like Schema	• Weakly Consistency • Availability first • Best effort • Approximate answer ok • Optimistic • Simpler and faster • Easy evolution

Fig 3 : ACID vs BASE [20]

3. Key-value and document oriented databases comparison

Key-value databases are used to store as schema-less data [9]. It stores data in form of associative arrays of entries Associative array is sometimes called as a map or symbol table or dictionary and sometimes also referred to as hash tables [10]. Associative array is a collection of (key, value) pairs string or the data is stored in any primitive datatype of any programming language or it is in the form of object. It is called as associative array as associative means 'binding' of data i.e. key and value together. 'Binding' is also known as creation of new association and other operation includes insert, remove, search, reassign. Most popular key value

stores are Amazon's Dynamo, MemcachedDB, Project Voldemort, Redis etc.

Document-Oriented databases are purposely designed for storing data in form of documents. Document –oriented databases uses an encoding schemes or file formats like XML, YAML, JSON (JavaScript object Notation), BSON (Binary JSON) etc [11]. Document –oriented databases are generally represented by a unique key where key can be a URI, path, or a string [11]. The idea behind designing this database is to add or delete any column from a particular row without creating dummy empty fields and this implies that we can add any number of column fields of any length [12]. Document oriented database also provides API or query language support to easily retrieval of documents.

Stephen Yen in its blog says "NoSQL is a Horseless Carriage" and suggested a classification on NoSQL's data model [22][23].Here we see the classification by Stephen Yen.

Data model	**Matching Databases**
Key-value Cache	Memcached, Hazelcast, Repcached, Coherence, Infinispan, EXtreme Scale, Jboss Cache, Velocity,Terracoqa
Key-Value Store	Keyspace, Flare, SchemaFree, RAMCloud
Key-Value Store - Eventually consistent	Dynamo,Voldemort, Dynomite, SubRecord, MotionDB,DovetailDB
Data-structures server	Redis
Key-Value Store – Ordered	Actord, Lightcloud, NMDB, Scalaris, TokyoTyrant,Luxio, MemcacheDB
Tuple Store	Apache River,Coord,Gigaspaces
Object Database	ZopeDB, db4o, shoal
Document store	CouchDB, MongoDB,MarkLogic, Jackrabbit, XML Databases, ThruDB, CloudKit, Perservere, Riak, Scalaris
Wide Columnar Store	BigTable, HBase, Cassandra, Hypertable, KAI, OpenNeptune, Qbase, KDI

Table 1: classification on data models [22][23]

Comparing Key-value and Document-oriented databases-

Database Name	MongoDB	CouchDB	Voldemort	Riak	MemcacheDB	DynamoDB
Developer	10gen	Apache software Foundation	Linkedin	Basho Technologies	Steve Chu	Amazon
Initial Release	2009	2005	2009	2010	2008	-
Major Users	Craiglist,Foursquare, shutterfly, Intuit	Lotsofwords.com	LinkedIn	Mozilla, Comcast, AOL	Facebook	Amazon
Storage type	Document	Document	Key-value	Key-value	Key-value	Key-value
Licence	AGPL, Open Source	Apache, Open Source	Apache, Open Source	Apache, Open Source, Proprietary	BSD, Open Source	Proprietary
Implementation Language	C++	Erlang	Java	Erlang	C, Python	Java
Best Use	Dynamic queries, frequently written, rarely read statistical data	Accumulating, occasionally changing data with predefined queries	-	High Availability	Small pieces of data with many concurrent connections. Transient data	large to big db solution
Key Points	Retains some properties of SQL such as query and index	DB consistency, easy to use	Data automatically replicated & partitioned to multiple servers	A truly fault-tolerant system, Riak has no single point of	Emphasis on persistence	-

				failure		
Concurrency Control	Locks	MVCC(Multi version Concurrency control)	MVCC(Multi version Concurrency control)	-	-	ACID
Replication	Async	Async	Async	Async	-	Sync
Deployment Model (On Premise)	Yes	yes	No	Yes	No	No
Platform	Mac OS X, Windows, Linux	Mac OS X, Windows, Linux	Mac OS X, Windows, Linux	Mac OS X, Unix, Linux	Mac OS X, Windows, Linux	Linux
Data Storage 1- BDB 2 –Disk 3- Plug-in 4- RAM	2	2	1,4	3	-	2
Characteristics 1-Consistency 2 - High - Availability 3 -Partition Tolerance 4-Persistence	1,3,4	2,3,4	2,3,4	2,3,4	1,3	1,2
Persistence Design	B-tree	Append only B-tree	Pluggable(primary BDB Mysql)	-	No	No
Query API	Cursor	Map/reduce views	Get/put	Nested hashes	-	-

Table2- Comparison between Key-value and Document-oriented databases [24][25][21]

CONCLUSION:

Section 2 analyse, describes and compare about RDBMS & NoSQL's databases. Section 3 contains information about key-value databases and document oriented databases. And their comparison. After analysing and discussion we conclude that, though NoSQL is an emerging, fast, highly scalable database which is used by big companies but still it has some drawbacks or we can say that it need some shortcomings like transactional support which stops capturing market from large section like banking sector which depends fully on transaction support. As NoSQL main objective is to achieve high availability and less on consistency and Relational databases still good for handling relations and consistency requirement. Although NoSQL was designed to have achieve zero-admin solution but still it requires skills to install and manage such databases. There are millions of RDBMS experts available but if we talk about NoSQL currently every developer is in learning stage. So finding experts for support in this field of this database is little difficult.

REFERENCES

1. http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/nosql/Home%20Page
2. G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's Highly Available Key-Value Store," in Proceedings of 21st ACM SIGOPS Symposium on Operating Systems Principles. New York, NY, USA: ACM, 2007, pp. 205–220.
3. B. Fitzpatrick, "Distributed caching with Memcached," Linux Journal, vol. 2004, no. 124, p. 5, August 2004.
4. J. Petrovic, "Using Memcached for Data Distribution in Industrial Environment," in Proceedings of the 3rd International Conference on Systems. Washington, DC, USA: IEEE Computer Society, 2008, pp. 368–372.
5. Mateusz Berezecki Facebook mateuszb@fb.com, Eitan Frachtenberg Facebook etc@fb.com , Mike Paleczny Facebook mpal@fb.com and Kenneth Steele Tilera ken@tilera.com "Many-Core Key-Value Store" http://gigaom2.files.wordpress.com/2011/07/facebook-tilera-whitepaper.pdf
6. http://en.wikipedia.org/wiki/NoSQL
7. http://www.tutorialindustry.com/nosql-tutorial-for-beginners
8. http://www.google.com/trends/explore#q=NOSQL&cmpt=q
9. Marc Seeger (21 September 2009) "Key –value Store:a practical overview" http://blog.marc-seeger.de/2009/09/21/key-value-store-a-practical-overview/ "Marc Seeger. Retrieved 1 January 2012"
10. http://en.wikipedia.org/wiki/Key-value_store
11. http://en.wikipedia.org/wiki/Document-oriented_database
12. Lith, Adam; Jakob Mattson (2010). "Investigating storage solutions for large data: A comparison of well performing and scalable data storage solutions for real time extraction and batch insertion of data" (PDF). Göteborg: Department of Computer Science and Engineering, Chalmers University of Technology. p. 70. Retrieved 12 May 2011
13. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.184.483&rep=rep1&type=pdf "Analysis and Classication of

NoSQL Databases and Evaluation of their Ability to Replace an Object-relational Persistence Layer" author: Kai Orend
14. Brewer, E. A. 2000. Towards robust distributed systems (abstract).In Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing (Portland, Oregon, United States, July 16 - 19, 2000). PODC '00. ACM, New York, NY, 7. DOI= http://doi.acm.org/10.1145/343477.343502
15. Gilbert, S. and Lynch, N. 2002. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. SIGACT News 33, 2 (Jun. 2002), 51-59. DOI=http://doi.acm.org/10.1145/564585.564601
16. Pritchett, D. 2008. BASE: An Acid Alternative.Queue 6, 3 (May. 2008), 48-55. DOI=http://doi.acm.org/10.1145/1394127.1394128
17. " Introduction to store data in Redis, a persistent and fast key-value database" Matti Paksula ,Department of Computer Science, University of Helsinki http://www.cs.petrsu.ru/fdpw/2010/article/paksula.pdf
18. http://en.wikipedia.org/wiki/CAP_theorem
19. "Joshi_Graph Visualization Using the NoSQL Database" http://library.ndsu.edu/repository/handle/10365/22972
20. http://www.yeeach.com/post/583"ACID vs BASE"
21. http://en.wikipedia.org/wiki/Memcached
22. http://en.wikipedia.org/wiki/NoSQL
23. A Yes for NoSQL Taxonomy. High Scalability (2009-11-05). Retrieved on 2013-09-18.
24. http://nosql.findthebest.com/
25. http://www.christof-strauch.de/nosqldbs.pdf