

Role of NoSQL in Big Data and Cloud Computing

¹Dr.M.Neelakantappa, Professor-IT

²Dr.Amjan Shaik, ³Dr.J.Sreedhar, Professor-CSE BVRIT Narsapur, TS, India
m.neelakanta@gmail.com

Abstract – This paper explores the scope and significance of NoSQL in data storage technologies. The gist of the data storage factors and compatibility of the NoSQL towards these factors are discussed in this paper, which also include the exploration of Big Data and role of NoSQL in Cloud Computing. The paper finally delivered the statistics of the rapid growth in data and the need of revolutionary strategies in data storage.

Keywords – NoSQL, NewSQL, Big data, Cloud Computing, Distributed Storage, Data Management.

I. INTRODUCTION

Reasons to Look for an Alternative to RDBMS

A significant long stride of Relational databases as data storage for responsible bequest applications of current business and operational transactions is getting slow down, which is due to the reasons claimed by development teams involving in software engineering process. As per these claims, the RDBMS models are not meeting the current needs, and those are following:

- The new formats of the recent data, such as ordered, unordered, partially ordered and unrelated structure data sets.
- Frequent data pushing.
- Need of flexible and dynamic data formats to compatible to modern programming.
- The vast usage of the scale out architecture by the Organizations under cloud and commodity servers.

NoSQL alternative to RDBMS

NoSQL represents Not Only SQL and the key standards of NoSQL environment to adopt as an alternative to the RDBMS are:

- Capable to manage High velocity of input data from divergent locations.
- Capable to store all formats of data such as ordered, partially ordered, unordered and unrelated structure datasets
- Capable to balance the volume of data in size of terabytes and petabytes.
- Capable to manage data complexity due to the storage at divergent locations and data centers.

NoSQL Database

NoSQL is a non-relational and widely distributed database environment that makes fast informal collection and analysis of data. NoSQL database was developed to meet the requirements of data rapidly increasing in quantity, intricacy, storage format and for its functionalities like speed of access and operational efficiency. This ever increasing huge data that is varied

and distinct led to the development of various variations of NoSQL and is identified as non-relational databases, cloud databases, Big Data databases among many other terminologies used.

Objectives of NoSQL Database

- NoSQL databases are designed to expand transparently by scaling out to low-cost commodity nodes.
- Due to magnitude growth in transaction rates over the last decade leads to massive increment in data storage, in regard to this NoSQL handles the volumes of "big data".
- The NoSQL architecture demands less management, hence it minimizes the Database Administrating role due to its capabilities of automated recover and repair, distributing the data, and simplified models of the data.
- Due to the usage of minimal cost commodity nodes as data storage clusters, the NoSQL reflecting drastic down fall of data storage cost and data process cost.
- NoSQL propelled no restrictions regarding data models; hence it allows any kind of structure of data to be saved.

NoSQL Database Constraints

- Many of available NoSQL databases are in early state of adaptations with lack of considerable characteristics. The scope of development comfort claimed by NoSQL, most of the developers attract to this as a significant alternative to RDBMS, but enterprise owners of the data should be cautious to select NoSQL as an alternative to RDBMS.
- Majority of the benchmarking NoSQL systems available are open source, hence the support is limited and expecting contributions from industry legend enterprises.
- Programming expertise is an essential requirement even in simple queries, which is due to the unavailability of business intelligence tools to connect to the NoSQL.
- Though the design standards are ignoring the role of administration, the current NoSQL adaptations demands the skilled professionals deploy and manage.
- Expertise in NoSQL usage is not common as like as RDBMS usage, which is due to the early state of the NoSQL Models. Hence the lapses of development time bounds and over loaded developer schedules are quite often. But this constraint is minimizing with magnitude speed, which is due to the simplification of NoSQL usage in current versions.

International Conference on Advances in Engineering Management & Sciences - ICEMS -2017

II. DATA MODELS OF NoSQL

- Document model: This model pairs each key with a polymorphic structure called document. This polymorphic structure contains lot of divergent key-value pairs. Here in these pairs a value is a scalar value, a vector of values or another polymorphic structure.
- Graph model: This model is specific to maintain the users information and their relations in social networks.
- Pairing each item with a key is referred as Key-value model, which is an easy and light weight NoSQL database. In addition to simply storing items with a key as identifier, stores the type of the value also.
- Wide Column models: These are specific to querying extremely large datasets. In regard to this, these models store the data as packed columns (more than one column data will concatenate as one entity

The Possible Advantages of Opting NoSQL Dynamic Schemas

In current agile development scenario, the format of the data is not stable and changes frequently with small snippets of the data, hence the fixed schema structure that used in RDBMS is not at all feasible in this regard. Hence the NoSQL is not supporting fixed schemas, rather, it supports the schemas used here can be changed dynamically. As NoSQL databases allow the insertion of data without a predefined schema, which makes it simple to adapt changes to applications and delivers no admin based reliable and fast transaction scope.

Auto-sharding

Expanding the storage area of a database across the multiple servers is called sharding. The complexity of sharding in traditional RDBMS is that the making hardware of divergent servers acts as a single server, since RDBMS shall not allow the expansion of database on divergent hardware environment. This Sharding in RDBMS is manual and also limits the veracity of the transactions. In contrast to the sharding in RDBMS, the NoSQL sharding is automatic and can be done without explicit server pooling, since the NoSQL can scale-out to divergent compatible hardware. The sharding in NoSQL is also capable to balance the load of data and queries over server pool automatically and also capable to manage any of server downtime without application interruption. Hence it is possible to use set of commodity servers as hardware platform as a single high-end server for a negligible cost that compared to sharding in RDBMS.

Replication

Structurally replicating the data leads to high availability and recovery against failures, Henceforth the NoSQL enables automatic replication with no explicit efforts and administration.

Inline Caching

Transaction caching is an excellent process that improves system performance; The RDBMS models are equipped with a caching mechanism that caches only overloaded read transactions towards improving system reading transactions performance. Due to its security enabled complex architecture, the RDBMS is not having the caching facility for write transactions, even unable to apply caching process on transactions that are mix of read and write. In contrast to the caching process of RDBMS, The NoSQL has excellent inline caching strategy that overcomes all the RDBMS limits explored.

III. THE INFLUENTIAL FACTORS OF MIGRATING TOWARDS NoSQL

The wide applicability of NoSQL compared to relational databases is because of multiple factors and mostly due to;

- 1) Support to random storage and indexing of Big Data sets serving a huge user base;
- 2) Quick reading and writing of data;
- 3) Mass storage capability;
- 4) Scalability;
- 5) Cost reduction in terms of storage and processing data;

Big Data

Data today is generated in huge volume and is dynamically changing overtime comprising of user-generated data or their applications machine-generated data. This includes user related personal information, system logging data, location dependent data, graph oriented data, real time generated data etc. This data can be quickly and effortlessly aggregated and made available for companies such as Facebook, D&B and Twitter. The data analyzed so is being used to build new applications or enhance existing ones and is transforming the way user's interact with the web in terms of voice, email or message communication, entertainment, social media, ads and e-commerce.

So a database should be flexible to handle, build efficiently a data set, store and retrieving it quickly using any type of data, semi-structured, unstructured or various other data types.

Semi structured and unstructured data are limited by the inflexibilities of relational database such as the rigid schema strategy. However NoSQL meets the growing user requirements with a data model that is devoid of such schema, improves application and database communication and reduces code redundancy, recoding and maintenance issues.

Big Users

The internet connects two billion people globally and the average time spent online by a user is increasing, and at the same time the number of concurrent users using an application is also gradually on the rise. Till recently an application with thousands of users was considered a huge

International Conference on Advances in Engineering Management & Sciences - ICEMS -2017

number whereas many applications today deployed on the cloud are available 24/7 and 365 days a year and have a large and varied user base that is more than a million users daily.

An application priority should be to support dynamically the ever increasing user base that logs in simultaneously. However due to the varied and dynamic nature of the number of users accessing the application at any point of time, the task to accurately foresee the number of concurrent users has attained great significance. Thus the main factors to be considered for any application to dynamically support its increasing user base accessing the application at a time are;

- Support to a user base ranging from zero to millions should one of the main features of an application.
- Support should not be limited to global users who access the application regularly but also to users who use the application infrequently.
- Scalability at given time providing seamless interaction to the user should define every new application development.

The inefficiency of relational database technologies in terms of scalability and quick access time is causing developers to opt for NoSQL and its increasing relevance.

Cloud Computing

Cloud computing services with distinct features is a platform that provides a scalable function over the internet where the host is a physical or virtual host. Till recently many services used two-tier client host architecture where the host is the service provider and client is the user with an individual system and the user base was fixed. Now a cloud service provider uses a three tier internet architecture run on a private or public cloud supporting a huge number of contingency users globally.

In three-tier architecture a web browser or a mobile application is used by the user to access the application over the internet. The cloud employs a scale-out model to support sudden increase in global concurrent users deploying further resources like servers such as an asset server to the web/application tier where a load balancer manages the increasing volumes of traffic.

However relational databases are inefficient with rigid features and fail in scale-out performance that is overcome using NoSQL database whose scale-out performance is excellent using the cloud services and three-tier internet architecture.

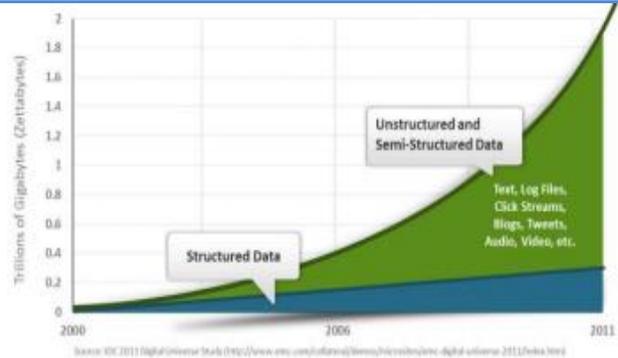


Fig. 1. The rapid growth of data in volumes and the drastic changes in structure of data; the Big Data [Source: www.couchbase.com/why-nosql/nosqldatabase]

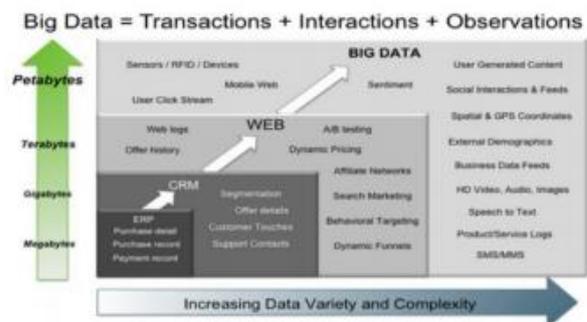


Fig. 2. Representation of the magnitude speed observed about transactions in Big Data [Source: <http://hortonworks.com/blog/7-key-drivers-for-the-big-data-market>]



Fig. 3. The rapid growth in the users, which is due to the usage of smart phones and the apps available [Source: www.couchbase.com/why-nosql/nosqldatabase]

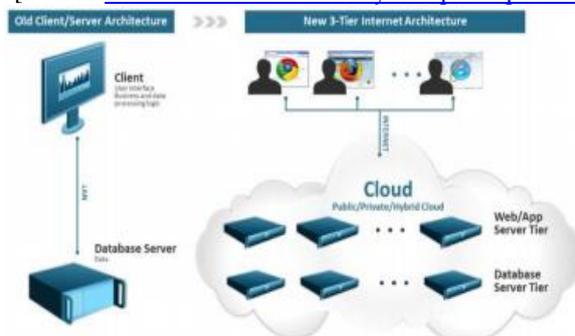


Fig. 4. The 3 layered architecture of internet, which is in the scope of scalable data process against magnitude speed of user’s growth

[Source: www.couchbase.com/why-nosql/nosqldatabase]

International Conference on Advances in Engineering Management & Sciences - ICEMS -2017

IV. CONCLUSION

NoSQL database is an emerging database technology and if further developed according to the criteria defined can give great results. Enterprises using the database should understand the various issues surrounding the final viability in terms of concerns such as user security and privacy associated with the data.

ACKNOWLEDGMENT

We would thankful to all the authors in the reference list for making this paper in a brighter way.

REFERENCES

- [1] Facebook Newsroom A New data center for Iowa. <http://newsroom.fb.com/News/606/A-New-Data-Center-for-Iowa>. Accessed 29 Sep 2013
- [2] Ohlhorst FJ (2013) Big Data Analytics: Turning Big Data into Big Money. John Wiley & Sons, Inc, Hoboken, New Jersey, USA
- [3] Beyer MA, Laney D (2012) The Importance of “Big Data”: A Definition. <http://www.gartner.com/id=2057415>. Accessed 29 Sep 2013
- [4] Agrawal D, Das S, El Abbadi A (2011) Big data and cloud computing: Current State and Future Opportunities. Proceedings of the 14th International Conference on Extending Database Technology - EDBT/ ICDT’11. ACM Press, New York, NY, USA, pp 530–533
- [5] Bughin J, Chui M, Manyika J (2010) Clouds, big data, and smart assets: Ten tech-enabled business trends to watch. McKinsey Quarterly 2010:1–14
- [6] Mell P, Grance T (2011) The NIST definition of cloud computing. NIST special publication 800–145. <http://csrc.nist.gov/publications/mistpubs/800-145/SP800-145.pdf>. Accessed on 29 Sep 2013
- [7] Zhang Q, Cheng L, Boutaba R (2010) Cloud computing: state-of-the-art and research challenges. J Intern Serv Appl 1:7–18. 10.1007/s13174-010-0007-6
- [8] Venters W, Whitley EA (2012) A critical review of cloud computing: researching desires and realities. J Info Technol 27:179–197. 10.1057/jit.2012.17
- [9] Tudorica BG, Bucur C (2011) A comparison between several NoSQL databases with comments and notes. 2011 10th International Conference RoEduNet. IEEE:1–5
- [10] Hecht R, Jablonski S (2011) NoSQL evaluation: A use case oriented survey. Proc 2011 Int Conf Cloud Serv Computing:336–341
- [11] Cattell R (2011) Scalable SQL and NoSQL Data Stores. ACM SIGMOD Record 39(4):12–27
- [12] Pokorny J (2011) NoSQL Databases: a step to database scalability in Web environment. Int J Web Info Syst 9(1):69–82
- [13] Sakr S, Liu A, Batista DM, Alomari M (2011) A survey of large scale data management approaches in cloud environments. IEEE Commun Surv Tutorials 13(3):311–336
- [14] Sadalage PJ, Fowler M (2013) NoSQL distilled: a brief guide to the emerging world of polyglot persistence. Addison-Wesley, Upper Saddle River, NJ
- [15] Abiteboul S, Manolescu I, Rigaux P, Rousset M-C, Senellart P (2012) Web Data Management. Cambridge University Press, New York
- [16] Aslett M (2011) How will the database incumbents respond to NoSQL and NewSQL? <https://451research.com/report-short?entityId=66963>. Accessed 29 Sep 2013
- [17] Lakshman A, Malik P (2010) Cassandra: a decentralized structured storage system. ACM SIGOPS Operating Syst Rev 44(2):35–40. 10.1145/1773912.1773922
- [18] Brewer E (2012) CAP twelve years later: How the “rules” have changed. Computer 45:23–29. 10.1109/MC.2012.37
- [19] NOSQL meetup. Eventbrite, San Francisco. <http://nosql.eventbrite.com/>. Accessed 29 Sep 2013
- [20] Konstantinou I, Angelou E, Boumpouka C, Tsoumakos D, Koziris N (2011) On the elasticity of NoSQL databases over cloud management platforms. Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM ’11. ACM Press, New York, NY, USA, pp 2385–2388
- [21] Apache CouchDB. <http://couchdb.apache.org/>. Accessed 29 Sep 2013
- [22] Corbett JC, Dean J, Epstein M, Fikes A, Frost C, Furman JJ, Ghemawat S, Gubarev A, Heiser C, Hochschild P, Hsieh W, Kanthak S, Kogan E, Li H, Lloyd A, Melnik S, Mwaura D, Nagle D, Quinlan S, Rao R, Rolig L, Saito Y, Szymaniak M, Taylor C, Wang R, Woodford D (2012) Spanner: Google’s globally-distributed database. Osd12:1–14
- [23] VoltDB Inc (2013) VoltDB Technical Overview. 1–4. http://voldb.com/downloads/datasheets_collateral/technical_overview.pdf. Accessed 29 Sep 2013
- [24] Clustrix Inc (2012) A New Approach: Clustrix Sierra Database Engine. 1–10. http://www.clustrix.com/wp-content/uploads/2013/10/Clustrix_A-NewApproach_WhitePaper.pdf. Accessed 29 Sep 2013
- [25] NuoDB Greenbook Publication (2013) NuoDB Emergent Architecture. 1–20. http://go.nuodb.com/rs/nuodb/images/Greenbook_Final.pdf. Accessed 29 Sep 2013
- [26] DB-Engines Ranking. <http://db-engines.com/en/ranking>. Accessed 29 Sep 2013
- [27] MongoDB.. <http://www.mongodb.org/>. Accessed 29 Sep 2013
- [28] Couchbase Server The NoSQL document database. <http://www.couchbase.com/couchbase-server/overview>. Accessed 29 Sep 2013
- [29] Apache HBase. <http://hbase.apache.org/>. Accessed 29 Sep 2013
- [30] Redis. <http://redis.io/>. Accessed 29 Sep 2013
- [31] Memcached. <http://memcached.org/>. Accessed 29 Sep 2013
- [32] Klopheus R (2010) Riak Core: building distributed applications without shared state. Proceedings of CUF’10 - ACM SIGPLAN Commercial Users of Functional Programming. ACM Press, New York, NY, USA, p 1
- [33] Oracle Berkeley DB 12c. <http://www.oracle.com/technetwork/products/berkeleydb/overview/index.html>. Accessed 29 Sep 2013
- [34] Neo4j - What is a Graph Database? <http://www.neo4j.org/>. Accessed 29 Sep 2013
- [35] Auradkar A, Botev C, Das S, De Maagd D, Feinberg A, Ganti P, Gao L, Ghosh B, Gopalakrishna K, Harris B, Koshy J, Krawec K, Kreps J, Lu S, Nagaraj S