Big-Data NoSQL Databases: A Comparison and Analysis of "Big-Table", "DynamoDB", and "Cassandra"

Sultana Kalid, Ali Syed, Azeem Mohammad, Malka N. Halgamuge
School of Computing and Mathematics, Charles Sturt University, Melbourne, Victoria 3000, Australia
e-mail: ASyed@studygroup.com, AMohammad@studygroup.com, MHalgamuge@studygroup.com

Abstract— **The growth and enhancement of technology in the corporate society has led to data storage and confidentiality issues. The problem arises from the management of trillions of data, generated every second in corporations, precisely known as "Big Data". Big Data needs to be stored and managed by larger companies that do not have the right storage systems, as there is not any developed yet. The aim of this paper is to find a solution to this growing problem by analyzing gaps in the literature, and to evaluate possible solutions. This study has analyzed content from top reviewed scientific publications, to gather compared and contrasted data from articles and highlight gaps. The highlighted literature will address this problems, and find solutions by contrasting BigData management approaches of NoSQL databases; BigTable, DynamoDB, and Cassandra. The findings summarized from publications are highlighted and the main features of all three databases and their applications are displayed. The system performances are analyzed based on their consistency, availability and partition intolerance. The study concluded that Google's BigTable and Amazon's DynamoDB are also critical and efficient on their own, and also found that the combination of both systems had caused the development of Cassandra. Cassandra is now the primary focus of numerous companies to develop different applications. Furthermore, all three systems are NoSQL storage systems, and BigTable, and based on one master node approach, unlike Dynamo, and Cassandra, it follows a Peer-to-Peer system. BigTable however, with some additional features from DynamoDB has helped the development of Cassandra, which is the basis of various modern applications available both open source and socially.**

Keywords- Big Data, BigTable, DynamoDB, Cassandra, RDMS, CAP

I. INTRODUCTION

Data is generated and processed more rapidly than ever; in recent times as more than 2.5 trillions of data is generated on a daily basis. Data generation will continue to increase in volumes in the future at an exponential level. This gives an understanding of how useful the tool "Big Data" is in today's concept of data generation as it helps store and manages vast amounts of data that is produced every second, each day [1]. One key technological problem the world is facing in regards to data storage and management is that, over the last few decades the storage and management of millions of data is being generated at intervals of less than a nanosecond. Therefore, the management of large amounts of data is a serious challenge to manage. With the growth of population, there is a need for an advanced data collection and management technology. Although relational database management systems (RDMS) have existed for some years for insistent data storage, they were yet to tackle scalability, consistency, system efficiency, data collection and integration of data extraction. To manage this phenomenal increase in data generation and to address all those above-mentioned challenges, an alternative solution must be brought into practice. Some databases such as, "NoSQL databases" has been introduced [2] nonetheless it remains inefficient due to high volumes of data production.

Real-time Data processing is a demand in today's word, as the processing is required in such a way that it does not have an adverse effect on the system's efficiency. Real-time Data processing requires: (i) The processing of large amounts of data while sustaining a system's high performance, (ii) a high availability and tolerance to network partitioning, (iii) The simple storage access 'API', (iv) Distribute systems and replicate data to globally distribute them to data centres, (v) Requirements varying between high throughput and low latency [3]. In response to the mentioned problem of data storage issues, and with rise of technological enhancement, this study has analyzed content to highlight gaps in the literature by comparing and contrasting from 12 top-reviewed scientific publications. The purpose of this research is to outline a detailed analysis of Google's BigTable, Amazon's Dynamo and Apache's Cassandra by comparing and contrasting top reviewed articles. "Google" and "Amazon" use two different approaches of NoSQL (BigTable and Dynamo), nonetheless Cassandra is a combination of the main features involved in BigTable and Dynamo database. This study will give a correct understanding of how companies are currently managing BigData instead of the "Traditional Data Management" system; the paper will then follow with the discussion about Facebook inbox search feature that encourages the introduction of the Cassandra NoSQL database.

II. NOSQL DATABASES OVERVIEW

A. *Google's BigTable*

The BigTable approach is a column-oriented storage system for applications of high performance. Being a distributed storage system, BigTable can handle "Structured" data. Various requirements of systems like scalability, performance, and availability of the systems achieved by BigTable products and projects [4] need to be considered. Google has developed a NoSQL based data management system for internal use of their corporation, titled "BigTable". BigTable is a distributed storage system that manages "structured data". Big data scalability to serve for some

Google applications is availed from this system. Due to their use of "Multi-dimensional sorted Map", BigTable is very flexible regarding many aspects such as data model. Google File System (GFS) is the storage platform for BigTable, highly scalable [5] [6]. The files are divided into chunks (chubby files) that are replicated across multiple machines involved in the whole system. The purpose of this process is to increase the availability of records and reliability of the system on real-time processing. The key features are: (i) Distributed storage system for “Structured Data” (ii) Hierarchical namespace (iii) Its Sparse (iv) High scalability (v) Strong consistency (vi) Persistent (vii) Multi-dimensional and Sorted-map (viii) Map is indexed by a unique “row-key”, “column key” and “a time-stamp”.

B. *Amazon’s Dynamo Database*

Amazon is using Dynamo Databases, which is a fully managed NoSQL database that provides predictable and super-fast performance with unified scalability. This provides a fundamental storage platform within Amazon’s system. There are various services on Amazon’s platform that involves high-reliability requirements, and only requires “primary-key” access to data storage nonetheless it is a relational database. However, to have a limited scalability and choose consistency over availability may not be ideal. In contrast with RDMS, Dynamo is designed to be eventually consistent, nonetheless highly scalable. The key features are: (i) Peer-to-peer systems Structured and unstructured, (ii) Distributed file system, (iii) Scalable and decentralized, (iv) Support only key-value API (no hierarchical namespaces or relational schema), (v) Efficient Latencies (no multi-hops), (vi) Highly-available key-value storage system, (vii) Availability, consistency, performance, (viii) Data partitioned using consistent hashing, (ix) Consistency facilitated by object versioning, (x) Trusted Network, no authentication, (xi) Incremental scalability, (xii) Symmetry, (xiii) Heterogeneity, and Load distribution.

C. *Apache Cassandra Database*

Apache Cassandra is one of the vastly scalable NoSQL. The approach of Cassandra allows corporates to analyze data in a particular fashion due to its ability to tackle generated data of BigData. Open source Cassandra was established in 2009, for the big data management ability of big companies such as Google, Facebook, and Amazon. Recently, Cassandra is being used by some modern businesses to handle acute ‘Data Infrastructure'. Unlike BigTable and Dynamo, Cassandra is the first choice for professionals as it offers more flexibility regarding fault handling and managing wide range data. This is done without compromising on the performance of the system [7]. Cassandra uses a blend of techniques employed by other NoSQL databases such as Big table and DynamoDB. As a result, Cassandra can maintain availability and scalability. A good example of this is the recent issue was Facebook Inbox search problem, as Cassandra was designed specifically to resolve the issue by enabling the users to search through their account’s inbox folder. This was a challenge, as millions of their users write or search per minute, and the handling of throughput is required. Users are being assisted from geographically distributed data centres, and the data replications through data centres have enabled the search invisibilities downcast. According to a recent research that has look into surveys, showed a successful result and now it is being used in many other storage systems for different services of Facebook and Instagram [8]. The key functions of this system are: (i) Peer-to-peer systems structured and unstructured, (ii) Decentralized storage system, (iii) Symmetric system orientation, (iv) Efficient latencies, (v) Linear scalability, (vi) Map is indexed by a unique “row-key”, “column key”.

III METHODS AND MATERIAL

Real-time system requirements particular styles that are based on availability, scalability, durability performances of the databases that have been studied in depth. Google File System BigTable maintains "Hierarchical-namespace", unlike Amazon DynamoDB that uses Peer-to-Peer storage systems. GFS is being used for products and projects of Google Inc. The architecture contains a primary (master) server for handling data sets; these data sets are stored in slave servers in the form of chunks [9]. On the other hand, GFS master server is referred to as “Chubby” abstraction, due to its fault tolerant ability [9]. Amazon uses DynamoDB for various projects and by numerous gaming applications [2]. Amazon uses this storage system to store and read/retrieve user's information specifically about shopping carts. Different conflicts resolve different approaches that are used for handling updated conflicts and network partitions, and this allows DynamoDB to write/read operation at any time. Traditional RDMS mainly focuses on enhanced consistency of the system, nonetheless, it becomes possible at an expense of systems availability and scalability [10] [11]. As they only ensure and offer high consistency, these systems usually fail during the handling of network partitioning. Peer- to-Peer connection and “Gossip” based algorithm enables every node of the system to sustain the information of each node. It is a structured overlap with a max one-hop request routing. BigTable offers both data distribution and structured system, nonetheless it depends on its availability for a distributed file system [9][12]. Similar to BigTable, Cassandra is "Distributed Multi- Dimensional Map System" that is indexed by using a particular key. Cassandra's storage layer is a local File System unlike BigTable's distributed file system and Dynamo's Pluggable Database. Operations that occur under each key are atomic for each replication irrespective of the number of columns read/written [4]. Furthermore, column sets (column families) are oriented in the same way as BigTable [13]. The request for either to read or to write is the key, which grows and routes to any node in system cluster [7] [11].

TABLE I. COMPARISON AND CONTRAST OF BIGTABLE, DYNAMODB, AND CASSANDRA BASED ON DIFFERENT STUDIES

	NoSQL Database	*Database applicability*	*System Performance*	*Scalability*	*Availability*	*Data operation*
Ping Yeh et al. (2009) [14]	**BigTable**	Applicable for Google projects and products	Very high with efficient scans	High	Access to most current data sets	Reading/ Writing (row, column or timestamp)
David Pearson et al. (2012) [15]	**DynamoDB**	The Web, social, mobile apps	Provisioned throughput model	Massive & seamless scalability	Consistent	Key-access & disk-only writes
Purna Chary et al. (2015) [13]	**Cassandra**	Facebook, Instagram, Netflix, eBay, etc.	Very high	Linear scalability	Continuous	Reading/ Writing

TABLE II. THE COMPARISON AND CONTRAST OF BIGTABLE, DYNAMODB, AND CASSANDRA BASED ON DIFFERENT STUDIES

NoSQL Databases	**Data Sets (Real-time processing)**	**Data Partition (Technique)**	**Data Scalability**	**Failure Tolerance**	**API**	**Study**
BigTable	Structured data	Dynamically partitioned hash tables	High	YES	Scan, get, del, put	Chang et al. (2008) [9]
DynamoDB	Structured & Unstructured	Consistent Hashing	Incremental	NO	Put, get	Weintraub et al. (2014) [6]
Cassandra	Structured & Unstructured	Consistent Hashing	Linear	YES	Get, put	A.Kashlev. (2015) [8]

TABLE III. CAP PROPERTIES OF BIGTABLE, DYNAMODB, AND CASSANDRA

CAP theorem	
BigTable	Consistency (**C**)
	Partition Tolerance (**P**)
DynamoDB	Accuracy (**A**)
	Partition Tolerance **(P)**
Cassandra	Accuracy (**A**)
	Partition Tolerance (**P**)

CAP THEOREM

Theorem gives information about the desirable features of RDMS [11] [16]. According to CAP theorem of each modern database can exhibit only two properties among Consistency (C), Availability (A) and Partition Tolerance (P).

IV. RESULTS & DISCUSSION

TABLE IV. HIGHLIGHTING KEY FEATURES OF BIGTABLE, DYNAMODB AND CASSANDRA

Features	**BigTable**	**DynamoDB**	**Cassandra (Daughter of Bigtable & DynamoDB)**
Data Management	Structured	Structured & unstructured	Structured &unstructured
Data Source	Strings, Graphs	Binary, string, number	All data types
Data Model	Multidimensional sorted map	Key value	Column family (BigTable)
System Orientation	Sparse	Symmetric	Symmetric
Data Storage	Column stores (SSTables)	Binary-value objects	SSTable Disk Storage
Load distribution	Load Balancing	Load sharing (heterogeneous)	--
Scalability	High	Incremental	Linear scalability
Consistency	High	Eventual	Eventual (DynamoDB)
Data Control	Dynamic	Tight	Yes
Language	JACOB	Java, .Net, Javascript, Perl, C#	JAVA
System isolation	No	Yes	No
Fault tolerance	Yes	No	Yes
Owner	Google Inc.	Amazon	Apache (Facebook, Instagram)
Platform	Linux, MAC, Windows	EC2	Linux, Nix
License	SaaS	SaaS	Apache license 2

This study has outlined a detailed discussion based on data sourced from reviewed articles and analyzed current data storage systems in BigTable, DynamoDB, and Cassandra. BigTable is a distributed storage system that can manage “Structured data”, and it is a column that is an oriented database system. Google is using this NoSQL database approach for its products and projects like Google File system, Google Earth and Personnel Search [2]. As discussed Amazon is a highly available system that uses ‘key-value’ to access the storage's "Structured" and "Unstructured" data. Amazon provides a platform for "e-commerce" that is used by millions of servers and customers distributed all around the globe [4]. This research [19] [22] has found that DynamoDB is a better approach while managing BigData, is a more reliable option for Amazon’s DynamoDB and BigTable. Data replication is on every node of the system, unlike BigTable, which includes one master node. This enables the system to avoid fault and unavailability. Our study has found that some enhancements in BigTable’s system can help a number of fields and applications. It has more flexibility to handle and manage BigData.

TABLE V. HIGHLIGHTING KEY APPLICATIONS OF BIGTABLE, DYNAMODB, AND CASSANDRA

Applications	
BigTable	
Fay Chang et al. (2008) [9]	Google File System
	Google Earth
	Personnel Search
DynamoDB	
Jeff Barr et al. (2016) [10]	Amazon S3
	Amazon SQS
	Amazon SNS
	TOKU HANDS
Nate Wiger et al. (2014) [6]	Battle Camp (game)
Daniela Miao et al. (2015) [12]	Mars Rover (NASA)
Apache Cassandra	
Ellis et al. (2014) [18]	Facebook
Ellis et al. (2014) [18]	Instagram

The arranged data in a distributed map and indexed search, show that [7][8][11], this approach is in the form of Cassandra. Although Cassandra is a combination of BigTable and DynamoDB, they are mostly a system orientation and other features that Google’s BigTable’s properties have. Cassandra offers the customers a simple model that allows a dynamic control over the data format and layout. It was firstly introduced for Facebook inbox search, nonetheless now numerous applications such as Instagram, Twitter, Rackspace, and gaming platforms are also used in this format. On top of this, investigating is a various security features [19][20][21] that could be an interesting avenue to explore in the future to protect BigData [22].

V CONCLUSION

Our study has reviewed databases based on top-reviewed scientific publications from 2010 to 2016 and has found that NoSQL, BigTable, DynamoDB, and Cassandra are applicable when managing big data. The contrast and comparison of these data's also show and demonstrate that the distinct features of these systems are potentially applicable to these systems from open sources. The observations based on the data are from the reported studies that clearly show that Cassandra is mainly based on BigTable's system architecture and data model. Key features of BigTable indicate that some modifications provided on the basis for the development of Cassandra would be ideal to use when managing big data. Cassandra was firstly used by Facebook inbox search, however, it is now being utilized by some social and private applications. In comparison with the hypothetical, interpretations of other studies that do not show all three systems that are non-relational storage systems. BigTable is based on one master node approach, unlike BigTable, Dynamo, and Cassandra, follows Peer-to-Peer system. BigTable however with some additional features from DynamoDB has helped in the development of Cassandra which is the basis of various modern applications available both open source, and socially.

REFERENCES

[1] G. Weintraub, “Dynamo and BigTable — Review and comparison”, IEEE 28th Convention of Electrical & Electronics Engineers in Israel (IEEEI), 5(4), 133-140, 2014.

[2] L. Wang and J. Zhan, J. “BigDataBench: A big data benchmark suite from internet services”, *IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*, 2014.

[3] P. C. Neves, “Big Data in the Cloud: A Survey”, OJBD, vol 1, no. 2, pp. 1-18, 2015.

[4] A. K. Kala, “*BigTable, Dynamo & Cassandra – A Review*”, International Journal of Electronics and Computer Science Engineering, 2(1) pp. 133-140, 2010.

[5] D. Pearson, “Amazon DynamoDB - Fast, Predictable, Highly-Scalable NoSQL Database”, 2012.

[6] G. Weintraub, “Dynamo and BigTable - Review and comparison”, IEEE 28th Convention of Electrical & Electronics Engineers in Israel (IEEEI), 1(2), 1-5, 2014.

[7] Datastax, Cassandra Summit, 2014.

[8] A. Kashlev, “KDM: An Automated Data Modeling Tool for Apache Cassandra, 2015.

[9] F. Chang, “Bigtable”. ACM Trans. Comput. Syst, 26(2), pp. 1-26, 2008.

[10] J. Barr, Amazon SQS | AWS Blog, 2016.

[11] A. Lakshman and P. Malik, “Cassandra” *ACM SIGOPS Operating Systems Review*, *44*(2), 35, 2010.

[12] M. Leppich, “Building a Mars Rover Application with DynamoDB”, 2015 Retrieved from http://www.infoq.com/articles/mars-rover-application-DynamoDB

[13] M. P. Chary and S. Kumar, “*A Survey on Implementation of Column-Oriented NoSQL Data Stores (Bigtable and Cassandra)”, 2015*.

[14] P. Yeh, “Distributed Database and BigTable”, Lecture Slides, Google, 2009.

[15] A. B. Moniruzzaman and S. A. Hossain, “NoSQL Database: New Era of Databases for Big data Analytics-Classification, Characteristics, and Comparison” IJDTA, 6(4), pp.1-14, 2013.

[16] F. Heinze, “DynamoDB vs. BigTable comparison | vs Chart.com.

[17] S. Ramanathan and S. Goel, “Comparison of Cloud database: Amazon's SimpleDB and Google's BigTable”, International Conference on Recent Trends in Information Systems, 1(2), pp. 165-168, 2011.

[18] J. Ellis, Jbellis presentations, 2014.

[19] D. V. Pham, A. Syed, A. Mohammad and M. N. Halgamuge, "Threat Analysis of Portable Hack Tools from USB Storage Devices and Protection Solutions", International Conference on Information and Emerging Technologies, pp. 1-5, Karachi, Pakistan, 14-16 June 2010.

[20] D. V. Pham, A. Syed, and M. N. Halgamuge, “Universal serial bus based software attacks and protection solutions”, Digital Investigation 7 (3), pp. 172-184, 2011.

[21] D. V. Pham, M. N. Halgamuge, A. Syed and P. Mendis, “Optimizing windows security features to block malware and hack tools on USB storage devices”, Progress in electromagnetics research symposium, pp. 350-355, 2010.

[22] V. Vargas, A. Syed, A. Mohammad, and M. N. Halgamuge, "Pentaho and Jaspersoft: A Comparative Study of Business Intelligence Open Source Tools Processing Big Data to Evaluate Performances", International Journal of Advanced Computer Science and Applications (IJACSA), vol 7, no 10, pp. 20-29, November 2016.