Integrating Hadoop and NoSQL Databases on AWS

Tamanna Baig
School of Computing
Clemson University
Clemson, South Carolina, USA
tbaig@clemson.edu

Vimal Vakeel
School of Computing
Clemson University
Clemson, South Carolina, USA
vvakeel@clemson.edu

***Abstract*—A study conducted by IBM in the year 2012 said that 2.5 exabytes of data is being created every day. Needless to say, we can expect this number to be significantly higher in the present context. The constant production of "Big Data" led to the creation of NoSQL databases which cater specifically to the volume of unstructured data that is being generated. Such databases focus primarily on unstructured data, faster read/write operations, and scalability. The analyses of such large volumes of data is of importance, not only in research, but also in the industry as well. Popular examples of such analytics being used are e-commerce and social networking websites which is mainly used for predictions and to formulate business decisions. Hadoop and NoSQL technologies both support a lot of common properties like varying data structures and data types. In this paper, we integrate two of the popular NoSQL database systems "MongoDB and "Cassandra" with "Hadoop" on an AWS cloud environment and observe its performance.**

***Index Terms*—NoSQL, Hadoop, MongoDB, Cassandra, Database schema, YCSB**

I. Introduction

Data in the internet can be broadly classified into four different dimensions which is called the 4V model of Big data [17]. Volume, velocity, variety and veracity of data is a direct result of the exponential growth of the internet. The source of such information is varies from news feeds, social networks, digital media, and scientific data sources, to name a few. The information generated by the web is not in a structured format [18] and cannot be represented into a two dimensional structure which is followed by traditional relational databases. Hence, NoSQL [19] databases were designed. The NoSQL databases are schema-free unlike relational databases which have a prerequisite to form the schema before storing the data in database. NoSQL databases can store unstructured data making them an appropriate tool for storage of big data. Various NoSQL databases are introduced in the market with each having its own specialty such as MongoDB [20], Cassandra [21], Redis [22] and Neo4j [23].

Traditional RDBMS systems are inefficient when it comes to scalability. A simple reason for this is that they were first conceptualized when the internet was still in its infancy and the applications were not expected to hold large volumes of data. Due to inefficiency of the relational databases "Facebook" had to use Memcache [24] to scale the data of user across the web. Both NoSQL databases, MongoDB and Cassandra, discussed in this paper fulfil the requirement of storing the unstructured information and the scaling of data as well.

Recently, NoSQL databases are being integrated with analytics tool so that they data can be analyzed with zero or minimal latency and further data can be used in making the managerial decision making. Instead for opting for the warehousing option where data is stored and the analysis on that data is done later, the companies are now opting for the real time analytics [25] so that they can know there customer well before their competition. Distributed data processing becomes important when the data itself comes in from various sources. We have used Hadoop [16] for this purpose. Hadoop is considered to be one of the best parallel processing tools which is open-source and inexpensive since it does require heavily configured clusters. The cluster configurations can be both homogeneous and heterogeneous.

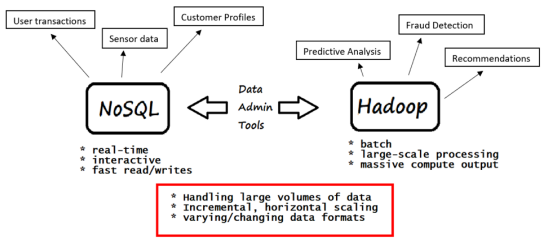

Fig. 1. Proposed design

Our novelty lies in the fact that we have used the high availability of an AWS Elastic cloud compute cluster to setup our Hadoop and NoSQL instances. AWS helps us to expand the ability to scale resources on-the-go but on a pay-as-you-go scenario [26] which adds to the inherent scalibilty that is available to us with the use of Hadoop and NoSQL.

II. Related Work

Although map-reduce and NoSQL technologies is largely used together in the industries but the comparative analysis of Hadoop and NoSQL integration for real time analytics is not very much researched upon topic. Abramova et al. [10] do comparison between the MongoDB and Cassandra without the hadoop attachment but they had done all the experiment on single virtual machine without considering

the scalability parameter which was the main reason for development of NoSQL databases. Govindaraju et al. [11] evaluates HDFS and MongoDB but they had done experiments on Cray XE6 machine having 153,216 "cores" which is used for the scientific purpose only and cannot be used in commercial activities by the regular companies and for the scalability purpose they had used the "cores" of computer but it cannot work in companies because they do not have the machine which have very large number of "cores" of this range. Cattell et al. [3] had examined the various RDBMS and NoSQL theoretically. Dory et al. [12] had compared the elastic scalability of Cassandra, MongoDB and HBase but without the mapreduce consideration. Dede et al. [13] evaluates the HDFS and Cassandra by considering the different parameters of Cassandra partitioning, scalability and replication factor and data locality. Padhy et al. [14] compare the data model and architecture of NoSQL technologies. Apart from MongoDB and Cassandra there are only few other NoSQL technologies which are designed to utilize the map-reduce framework of Hadoop such as HBase [15]. HBase which is developed by apache is also a column oriented database like Cassandra and it is shown that architecture changes of Hadoop and HBase can improve the storage and processing capabilities [16]. But when Cassandra and HBase is compared the Cassandra over powers the capabilities of HBase [17]. We have chosen both NoSQL technologies depending on market applicability, usability areas and rank among their respective database types. These studies do not directly represent the performance measurement between Hadoop-MongoDB and Hadoop-Cassandra. Rapid increase use of these technologies in real time analytics led us to evaluate their performance.

In this section, we present the characteristics of two NoSQL databases, Cassandra and MongoDB as well as the Hadoop and AWS environments.

III. Tools used

A. MongoDB

MongoDB is an open source NoSQL document oriented database system developed by MongoDB Inc. As already discussed in document oriented databases that data is stored in documents and not in tabular format despite all this MongoDB provides many features of relational databases which include indexing, replication, queries and provides much more improvement in load balancing, file storage. MongoDB also provides its own map-reduce features for the distributed processing of the data.

B. Cassandra

Cassandra is an open-source NoSQL distributed database [11] written in Java. It is an ideal fit to maintain huge amount of structured as well as unstructured data because of its ability to scale elastically as well as linearly. Because of Cassandra's linear elasticity, the performance increases with the increase in the number of nodes in the cluster. Easy data distribution by replicating data across several data-centers is supported by Cassandra. Cassandra also supports ACID properties like the relational databases and has very fast write speed. Cassandra falls into the category of being a Key-Value database because it is eventually consistent like Amazon's Dynamo [4] as well as a Column-Family database because it stores data in column families just like Google's Big Table [3].

C. Hadoop integration

Apache's Hadoop map-reduce framework is considered to be the best in the field of distributed computing. Whether if its for computational time, fault tolerance in a very large network, working on the cluster which are either heterogeneous or homogeneous and batch processing of data, it performance much better than other commercialy available resources. A minor drawback of the batch-processing is that it is not efficient for read-write operations on the Hadoop Distributed File systems (HDFS) because of which we need to use it with NoSQL databases [16].

D. Amazon Web Services - AWS

The AWS technology is implemented at server farms throughout the world, and maintained by the Amazon subsidiary. In aggregate, these cloud computing web services provide a set of primitive abstract technical infrastructure and distributed computing building blocks and tools [15]. One of these services is Amazon Elastic Compute Cloud, which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet.

IV. Experimental Setup

Amazon Web Services provides us various services that we can use on a pay-as-you-go basis. We have designed our experiment using Elastic Cloud Compute, or EC2, instances to setup the NoSQL databases and the Hadoop architecture.

The instances that we create have to be configured from the ground-up. This means that we have to select what type of compute resources we want, how much memory we need, what type of security settings do we need to setup in order to make sure the connections between the different instances are secure. For the purposes of this experiment, we have used a t2.micro EC2 instance. This instance has 8Gb of memory. These instances can be scaled up using different services such as auto scaling groups if we need to increase our compute power. After the memory setup is done, it is necessary to configure the security group details. The security group contains information that the instance needs to establish a connection with the remote computer(in our case, our local PC). Important information that it needs are the type of connection, TCP/IP, SSH. It also requires the port numbers and the IP addresses over which the communication should occur.

Once the instance is setup, we then need to download a ".pem" file that is generated after the instance is launched. Without this file, access to the instances will not be granted.

We had setup different instances for the NoSQL and Hadoop instances. Four instances were setup for the Hadoop cluster and 1 instance each for MongoDB and Cassandra.

Additional dependencies had to be configured before the NoSQL databases and the Hadoop instances could be installed. These dependencies included using Java sdk version 8, python version 2.7, and Maven 2.4.

Hadoop-3.2, MongoDB Server-4.2, and Cassandra-2.2 was then installed on the AWS instances. The servers for each of these were then started to check if the database was active or not after which the data was loaded onto the NoSQL databases.

4 Hadoop nodes had to be created configured to make sure one node served as a namenode(master) and three nodes as datanodes(slaves). The connections between MongoDb and Hadoop [27] and Cassandra and Hadoop [28] were then configured.

The YCSB benchmark was then installed and built and the expirement was run for different workloads as explained below.

V. Evaluation of Results

YCSB Benchmark client was used to analyse the performance of the proposed system under different scenarios that are relevant to real-time analytics. Yahoo! Cloud Serving Benchmark(YCSB) was used for the aforementioned experiments. Overall runtime was recorded for each workload by increasing the number of records exponentially each time. The experiment was repeated five times for each set of records and average of these timings was used to visualize the findings.

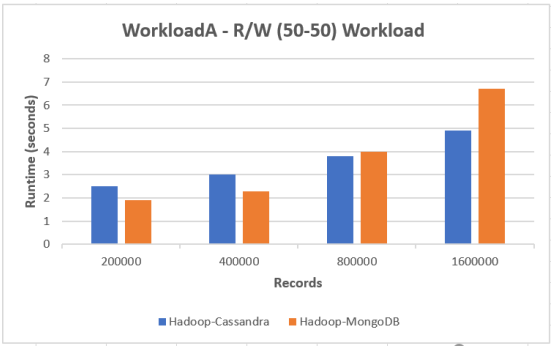

Fig. 2. Evaluation for 50-50 Read/Write transactions

A. WorkloadA: Balanced read/write transactions

This workload contains 50-50 distribution of both read and write transactions. The application of such an instance would be in the session store recording of recent actions. Figure 1 shows that when the number of records are relatively low, the Hadoop-Mongo system works comparatively faster than the Hadoop-Cassandra. But as the records are increased exponentially, the performance of Hadoop-Cassandra improves. We can conclude that as the number of records increases the efficiency of Hadoop-Mongo drops, on the other hand efficiency of Hadoop-Cassandra increases.

B. WorkloadB: Read-intensive transactions

This workload represents a read-intensive 95-5 distribution of read and write transactions. The application of such an instance would be in a typical social media website where the user is most-likely to retrieve large amounts of data to read rather than posting anything on their website. Figure 2 shows that when the number of records are relatively low, the Hadoop-Mongo system works comparatively faster than the Hadoop-Cassandra. But as the records are increased exponentially, the performance of Hadoop-Cassandra improves. Similar to WorkloadA, we can conclude that as the number of records increases the efficiency of Hadoop-Mongo drops, on the other hand efficiency of Hadoop-Cassandra increases.

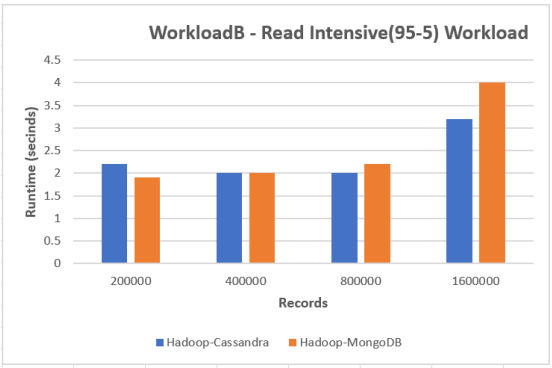

Fig. 3. Evaluation for Read Intensive transactions

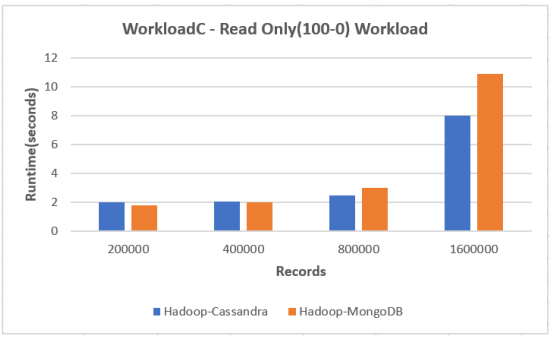

Fig. 4. Evaluation for Read Only transactions

C. WorkloadC: Read-only transactions

This workload represents an exclusive read-only distribution of transactions. The application of such an instance would be in Cache system. Figure 3 shows that when the number of records are relatively low, both the systems perform very closely to each other if not exactly equal. But as the records are increased exponentially, the performance of Hadoop-Cassandra improves. Similar to earlier workloads, we can conclude that as the number of records increases the efficiency of Hadoop-Mongo starts to drop heavily, on the other hand, efficiency of Hadoop-Cassandra increases.

VI. Comparing results to existing systems

The results obtained from the experiments were compared to an existing system available. This existing setup was setup by spawning Virtual Machines on local desktops, the results

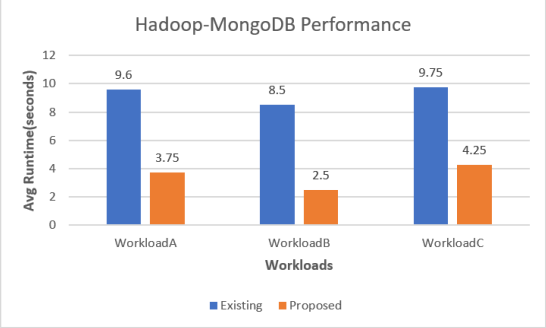

Fig. 5. Comparing to existing Hadoop-MongoDB system

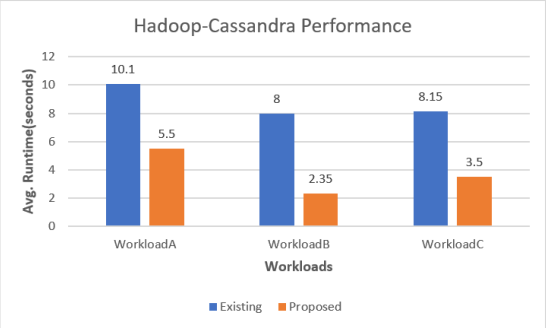

Fig. 6. Comparing to existing Hadoop-Cassandra system

are shown in Figure 4 and Figure 5. As depicted in the graphs, the proposed system works better in every way compared to the existing system. The main reason behind this would be the fact that our entire architecture is hosted on Amazon Web Services (AWS) cloud.

VII. Conclusion

It can be positively concluded that the proposed system works well computationally compared to the existing system. The idea of using NoSQL database for fast read/write transactions and Hadoop for large-scale analytics and integrating them would help a process exponentially.

Additional workloads could be run and their values evaluated for performance benchmarking to see if AWS provides better setup for NoSQL and Hadoop integration.

References

[1] April Reeve, Big Data and NoSQL: The Problem with Relational Databases, infocus.emc.com, September 2012.

[2] Brewer, E. (2012). CAP twelve years later: How the" rules" have changed. Computer, 45(2), 23-29..

[3] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., ... Vogels, W. (2007, October). Dynamo: amazon's highly available key-value store. In ACM SIGOPS Operating Systems Review (Vol. 41, No. 6, pp. 205-220). ACM.

[4] MongoDB Documentation, docs.mongodb.org

[5] Dharavath, R., Kumar, C. (2015). A scalable generic transaction model scenario for distributed NoSQL databases. Journal of Systems and Software, 101, 43-58.

[6] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[7] Leavitt, N. (2010). Will NoSQL databases live up to their promise?. Computer, 43(2), 12-14.

[8] Han, J., Haihong, E., Le, G., Du, J. (2011, October). Survey on NoSQL database. In Pervasive computing and applications (ICPCA), 2011 6th international conference on (pp. 363-366). IEEE.

[9] Cattell, R. (2011). Scalable SQL and NoSQL data stores. ACM SIGMOD Record, 39(4), 12-27.

[10] Wang, G., Tang, J. (2012, August). The nosql principles and basic application of cassandra model. In Computer Science Service System (CSSS), 2012 International Conference on (pp. 1332-1335). IEEE.

[11] Chodorow, K. (2013). MongoDB: the definitive guide. " O'Reilly Media, Inc."

[12] Pokorny, J. (2013). NoSQL databases: a step to database scalability in web environment. International Journal of Web Information Systems, 9(1), 69-82.

[13] Schwinger, W., Retschitzegger, W., Schauerhuber, A., Kappel, G., Wimmer, M., Pröll, B., ... Zhang, G. (2008). A survey on web modeling approaches for ubiquitous web applications. International Journal of Web Information Systems, 4(3), 234-305.

[14] Grolinger, K., Higashino, W. A., Tiwari, A., Capretz, M. A. (2013). Data management in cloud environments: NoSQL and NewSQL data stores. Journal of Cloud Computing: Advances, Systems and Applications, 2(1), 22.

[15] "AWS Global Infrastructure". December 22, 2016. Retrieved December 22, 2016.

[16] P.Z Cheng, J.J Ze, C.B Xioa, Z.K Zhi. Real-time analytics processing with MapReduce. International Conference on Machine Learning and Cybernetics (ICMLC), 2012, 4, 1308-1311.

[17] P. Hitzler, K. Janowicz. Linked Data, Big Data, and the 4th Paradigm. Semantic Web, 2013, 4, (3), 233-235.

[18] S.S Nyati, S. Pawar, R. Ingle. Performance evaluation of unstructured NoSql data over distributed framework. Advances in computing communications and informatics (ICACCI), 2013, 1623-1627.

[19] Cattell .R, Scalable SQL and NoSQL data stores. ACM SIGMOD Record, 2010, 39, (4), 12-27.

[20] A. Boicea, F. Radulescu, L.I Agapin. MongoDB vs Oracle Database Comparison. Third International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), 2012, 330-335.

[21] L. Avinash, M. Prashant. Cassandra - A Decentralized Structured Storage System. ACM SIGOPS Operating Systems, 2010, 44, (2), 35-40.

[22] Redis website. [Online]. Available from URL: http://www.http://redis.io/

[23] Neo4j website. [Online]. Available from URL: http://www.neo4j.org/

[24] N. Rajesh, F. Hans, G. Steven, K. Marc, L. Herman, L.C Harry, et al. Scaling Memcache at Facebook. In: Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation, 2013, 385-398.

[25] J. Rats G. Ernestsons. Using of cloud computing, clustering and document-oriented database for enterprise content management. 2013 Second International conference on Informatics and Applications, 2013, 72-76.

[26] "What is Cloud Computing by Amazon Web Services — AWS". Aws.amazon.com. Retrieved July 17, 2013.

[27] MapReduce-Usage website. [Online]. https://github.com/mongodb/mongo-hadoop/wiki/MapReduce-Usage

[28] Datastax website. [Online]. https://www.datastax.com/blog/2013/05/free-odbc-drivers-cassandra-and-hadoop-now-available