Imperial Journal of Interdisciplinary Research (IJIR)
Vol-3, Issue-4, 2017
ISSN: 2454-1362, http://www.onlinejournal.in

Comparative Performance Analysis of various NoSQL Databases: MongoDB, Cassandra and HBase on Yahoo Cloud Server

Rajith Kumar S[1] & R. Roseline Mary[2]
[1,2]Department of Computer Science, Christ University, Bangalore, India

***Abstract**: Databases provide data storage, extraction and manipulation by using SQL language. It has emerged as a backend to support Big Data applications. It is mainly characterized by horizontal scalability, schema-free data models, and easy cloud deployment. There are various NoSQL databases and the performance varies with different types based on node capacity, number of cores, replication factor, and different workloads. Hence, it is important to compare them in terms of their performance and verify how the performance is related to the different database. This paper focuses on comparison of Cassandra, MongoDB and HBase which are the most commonly used NoSQL databases. This comparison between NoSQL databases deploys them on yahoo cloud platform which uses different types of virtual machines and cluster sizes to study the effect of different configurations. The final result shows the performance of databases at different workload levels and the result can be compared to find out the best among these three databases. This comparison also allows users to choose the most appropriate database according to their needs.*

1. Introduction

Databases are considered as a vital part of the organization. It is being used all over the globe. Originally, relational database were used which helped in the storage, extraction and manipulation of large volumes of data. However, with the constant growth of data, relational databases have their own limitations. To overcome the limitations, a new database model was developed with additional features, known as NoSQL database (Non-relational database). These databases are much more efficient and were not limited to scalability and storage. NoSQL database emerged as a breakthrough technology and it is used as a complement to relational database.

NoSQL databases are used to maintain large amounts of data on cloud platforms mainly because of its horizontal scalability, schema free data models, and easy cloud deployment. These databases are widely used for big data applications. NoSQL database features horizontal scalability that is, the capacity to scale in performance when the number of machines added to an existing cluster increases. It is able to identify the correct number of nodes required for a specific workload. The experiments are conducted on fixed physical hardware architecture. The experiment is done to assess the performances of three different NoSQL systems i.e. Cassandra, MongoDB and HBase. This work is mainly aimed at characterizing the differences of NoSQL databases.

2. Cassandra:

Apache Cassandra in a nutshell is an open source, peer to peer distributed database architecture, decentralized, easily scalable, fault tolerant, highly available, eventually consistent, schema free, column oriented database. By contrast, Cassandra has a peer-to-peer distribution model, such that any given node is structurally identical to any other node that is, there is no “master” node that acts differently than a “slave” node. There is no need to store a value for individual column every time a new entity is stored. A cluster is a container for time signature spaces. A time signature space is the outermost container for data in Cassandra, but it’s perfectly fine to create as many key spaces as the application needs.

Partitioning defines how data will be distributed across the Cassandra nodes and allow you to specify how row keys should be sorted, which has a significant impact on the options available for querying ranges of rows. This has the advantage of spreading your keys evenly across your cluster, because the distribution is random. It has the disadvantage of causing inefficient range queries. Durability in Cassandra is ensured with the help of commit logs, which a crash recovery mechanism. Writes will not be considered successful until data is written to commit logs to support durability goals. When the number of objects stored in the table reaches a threshold, the contents of the memo table are flushed to disk in a file. Although Cassandra offers atomicity at the column family level, it does not guarantee isolation and no locks.

3. Mongo DB:

MongoDB is a flexible and scalable document oriented data store with dynamic schemas, auto sharing, built-in replication and high availability, full and flexible index support, rich queries, aggregation. Mongo DB follows a master-slave approach, and it has an automatic failover feature where if a master server goes down, MongoDB can automatically failover to a backup slave and promote the slave to a master. Master-slave replication is the most general replication mode supported by MongoDB, very flexible for backup, failover, read scaling. Shading is MongoDB's approach to scaling out. Shading allows you to add more machines to handle increasing load and data size horizontally without affecting your application. Shading refers to the process of splitting data up and storing different portions of the data on different machines. The basic concept behind MongoDB's shading is to break up collections into smaller chunks. These chunks can be distributed across shards so that each shard is responsible for a subset of total data set.

Write concern describes the guarantee that MongoDB provides when reporting on the success of a write operation. The strength of the write concerns determines the level of guarantee. When inserts, updates and deletes have a weak write concern, write operations return quickly. MongoDB provides different levels of write concern to better address the specific needs of applications. For a shared cluster MongoDB directs write operations from applications to the shards that are responsible for the specific portion of the data set. Without indexes, MongoDB must scan every document in a collection to select those documents that match the query statement. In MongoDB, these operations modify the data of a single collection. For the update and delete operations, criteria can be specified to select the documents to update or remove.

4. HBase:

HBase is called the Hadoop database because it is a NoSQL database that runs on top of Hadoop. It combines the scalability of Hadoop by running on the Hadoop Distributed File System (HDFS), with real-time data access as a key/value store and deep analytic capabilities of Map Reduce. This article introduces HBase and describes how it organizes and manages data and then demonstrates how to set up a local HBase environment and interact with data using the HBase shell.

Apache HBase is a NoSQL database that runs on top of Hadoop as a distributed and scalable big data store. This means that HBase can leverage the distributed processing paradigm of the Hadoop Distributed File System (HDFS) and benefit from Hadoop's MapReduce programming model. It is meant to host large tables with billions of rows with potentially millions of columns and run across a cluster of commodity hardware. But beyond its Hadoop roots, HBase is a powerful database in its own right that blends real-time query capabilities with the speed of a key/value store and offline or batch processing via MapReduce. In short, HBase allows you to query for individual records as well as derive aggregate analytic reports across a massive amount of data.

As a little bit of history, Google was faced with a challenging problem: How could it provide timely search results across the entire Internet? The answer was that it essentially needed to cache the Internet and define a new way to search that enormous cache quickly. It defined the following technologies for this purpose:

- Google File System: A scalable distributed file system for large distributed data-intensive applications
- Big Table: A distributed storage system for managing structured data that is designed to scale to a large size: petabytes of data across thousands of commodity servers
- MapReduce: A programming model and an associated implementation for processing and generating large data sets

5. Related study:

[i] Cassandra is a NoSQL database storage system for managing large amounts of data. It has high scalability and does not require high performance processing for higher input and output levels. It is very efficient with no single point of failure. Cassandra has the capability to handle very high write throughput, and is easily scalable with the number of users.

[ii] Cassandra, MongoDB and HBase are the three most popular databases. These are non-relational databases which are more efficient than relational databases and provide high scalability. These databases are tested for their performance based on number of nodes, number of cores and replication factor. The comparison between these databases shows how efficiently the storage system performs with the given workload.

[iii] The databases are evaluated on Yahoo! Cloud Serving Benchmark. The workloads are defined by

read and update. The databases are tested on three different workloads having 50% read and 50% update, 100% read and 0% update, 0% read and 100% update. After the evaluation results show that Cassandra's Performance was much faster than the HBase.

6. Experimental Setup:

Introduction on Yahoo Cloud Server: In this experiment Yahoo Cloud Serving Benchmark developed by Yahoo is used for testing the database performance. With YCSB, multiple databases can be tested and compared. The YCSB operates in two phases; initially the data's are loaded onto the data nodes. The data is generated randomly and stored by the database in its format. In the second phase random key requests are sent to the data nodes. The requests are mixed with 50% reads and 50% writes for each client thread. The number of client threads is either varied to simulate different workload levels, or fixed to a value that saturates the DB servers.

The test is performed using a specific DB configuration in terms of active cores per server, number of nodes and data replication. The YCSB tool is executed at least 20 times for each configuration before collecting the average values for response time and throughput.

7. Results and discussion:

Yahoo! Cloud Serving Benchmark is used to test and compare the performances of these NoSQL databases over different types of workloads. The performance of database is a primary factor for deciding the type of database required for use such as in an enterprise or for applications. It is necessary to compare them and analyses the difference of NoSQL databases, and provide a performance reference. The performances of the database were tested based on the effect of number of nodes, number of cores and replication factor. The databases were compared on a fair mode. Based on the above factors and test method the following results were established.

8. Number of Cores:

These tests are performed on a single machine of the c1.xlarge type for both Cassandra and MongoDB, in which each was configured with one node only. For HBase, the region server was hosted on a single machine of the c1.xlarge type, and the master and the zookeeper were each hosted on a different m1.xlarge VM.

Table 1

	No. of cores	K (ops/sec)
Cassandra	8	10
MongoDB	8	10
HBase	8	22

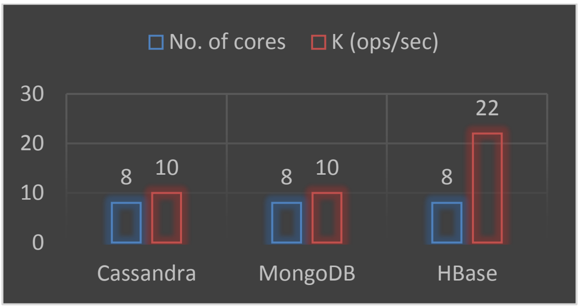

Fig 1 No.of Cores

The total throughput as function of the number of cores active on a single node is shown as in the table above. It is clear that HBase is able to take advantage of the number of cores better than the other databases. MongoDB scales well initially but then it shows a plateau at around 6 cores. This is not present in Cassandra which, although significantly slower that HBase in absolute terms, which scales quite linearly.

9. Number of Nodes:

To examine the effect of the number of nodes on performance the databases have been simulated on higher workload intensities by increasing the number threads running simultaneously on the client for each request.

Table 2

Cassandra		
No. of threads	Label	K (ops/sec)
30	1N/1_cpu	1
30	1N/2_cpu	3
30	2N/2EP/1REPL	5
30	4N/4EP/1REPL	6
30	8N/8EP/1REPL	13
30	16N/16EP/1REPL	14

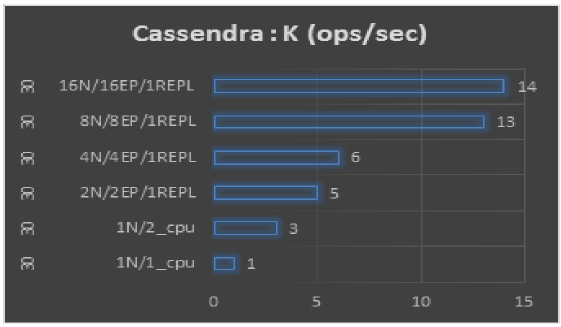

Fig 2 Cassendra No..of Nodes

Table 3

MongoDB		
No. of threads	Label	K (ops/sec)
30	1N/1_cpu	3
30	1N/2_cpu	5
30	2N/2_shard	7
30	4N/4_shard	13
60	8N/8_shard	24
60	16N/16_shard	34

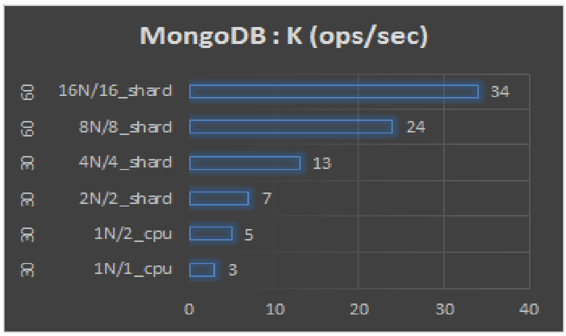

Fig 3 MongoDB No..of Nodes

Table 4

HBase		
No. of threads	Label	K (ops/sec)
30	1N/1_cpu	5
30	1N/2_cpu	8
30	2N	13
60	4N	28
60	8N	37

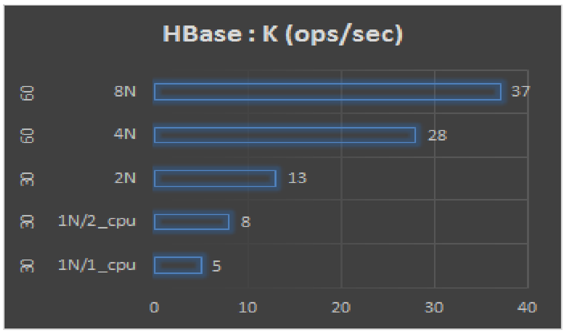

Fig 4 HBase No..of Nodes

10. Replication:

Replication of data in a DB cluster introduces yet another performance impacting factor. A replication factor equal to one suggests that there is only one copy of the data item among all the nodes of the cluster. Higher replication factors means that, aside from the original data, copies are stored on different nodes. The degradation increases as more copies of the same data are requested in order to achieve consistency within the database.

Table 5

	No. of nodes	Replication factor	K (ops/sec)
Cassandra	9	1	17
	9	2	14
	9	3	13
MongoDB	9	1	23
	8	2	12
	8	3	13
HBase	9	1	52
	9	2	33
	9	3	48

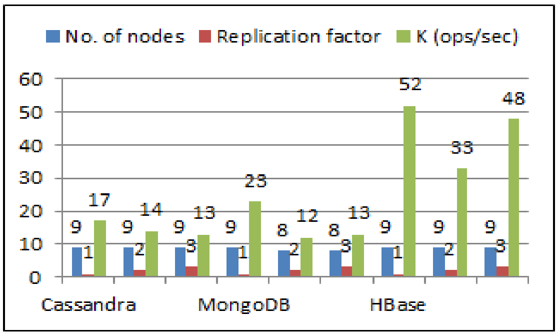

Fig 5 Replication

The above table shows the behavior of MongoDB when the replication factor is set to values higher than one. In this case, having replicas of data decreases the throughput, but differently from Cassandra, having more than one replica does not decrease performance. The effects of replication in HBase are shown. Because of the architecture of HBase, the behavior is not what was expected. There was no noted decrease in performance or in throughput or in response time. This is due to the fact that Hadoop is responsible for data allocation in a process transparent to the user. Moreover, HBase packs its data regions and splits data among its nodes randomly hence no uniform architecture for data distribution exists.

11. Conclusion:

The use of NoSQL databases and its popularity has increased as large amount of data are being processed. These databases have lot more advantages compared to relational databases, especially for large volumes of data. This research gives benchmarks and models for three of the most common NoSQL databases Cassandra, MongoDB and HBase. The databases were tested on the yahoo cloud platform using different types of virtual machines to study the effect of different configurations and illustrate the performance behavior of the databases. The results showed that the models are able to capture much of the main performance characteristics of the studied databases at high workloads. Among the three, Cassandra and HBase were found much better in performance than MongoDB.

12. References

[1] Avinash Lakshman and Prashant Malik, “Cassandra - A decentralized structured storage system”

[2] A. Gandini, M. Gribaudo, W.J Knottenbelt, R Osman, P. Piazzolla, “Performance evaluation of NoSQL databases”

[3] Veronica Abramova, George Bernardino, Pedro Furtado, “Which NoSQL database? A performance overview”