ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 6, Issue 2, February 2017

A VIEW ON LOAD BALANCING OF NOSQL DATABASES (COUCHBASE, CASSANDRA, NEO4J AND VOLDEMORT)

Dildar Husain M.Tech,
Department of CS&IT
Maulana Azad National Urdu University,
Hyderabad

Mr. Mohammad Omar,
Department of CS&IT,
Maulana Azad National Urdu University,
Hyderabad

***Abstract*—Websites like facebook and twitter has become the daily life of millions of people where they share photos, messages, tweets, likes, comments, tag, create posts and do many other things. So all these things need to be stored in a database where it can be accessed and modified with difficulty and with fast speed. For this problem we cannot use the relational databases as it will not be a good choice for this high amount of access and also because of the nature of data. This data can be unstructured so for this we use NoSql databases. Data is ever increasing and its increase rate is very high as the no of users increases to access the services so sometimes the server may not be able to handle all the requests so the concept of Load balancing comes so that the load from the server can be distributed to other sites so that services can be provided to all the users without any halt or discontinuation. Load balancing can be done in different nosql databases with different way.**

Keywords— NoSQL, couchbase, mapping function, Replication, rebalance, Neo4j, orchestrator, virtual bucket, water mark, voldmort.

1. Introduction

Before start the discussion about Load Balancing in NoSQL Database let's take a look at what is NoSQL database NoSQL is a fast, portable, relational database management system without limits (other than memory and processor speed) that run beneath, and interacts with, the UNIX Operating System. It uses the "Operator-Stream Paradigm"[1]. NoSQL can also be pronounced as "Not Only SQL" or "noseequence" which is a derivative of the RDB database system. In another way we can say that a NoSQL database environment is a non-relational as well as largely distributed database system that enables quick, ad-hoc organization and extremely high-volume, disparate data types. There are many NoSQL databases such as Document-Oriented Databases, In-Memory Databases, Graph Databases and Column Store Databases that differ from each other by their Architecture, Data Model, Data Distribution Model, and Development Model. [2] Servers can be integrated into a cluster or abstracted from a live cluster these servers are employed side-by-side with subsisting servers and place abaft a load balancer.

1.1 BALANCING IN DOCUMENT-ORIENTED DATABASES (COUCHBASE)

There are many types of Document Oriented Databases [2]

Load balancing is achieved in Couchbase database on server side mainly in two ways..

(I)There are Hash function keys which are distributed over the servers by the Virtual Bucket.[4]

On the client-side Couchbase Server the load is balance during the data operation by hash function key and server mapping algorithm among clusters. When an application utilizer takes an action which needs to update a data item in Couchbase Server. The application server respond to the utilizer action updates the key's value and makes a call to a Memcached client library to set the key-value pair. For transmitting the operation to the server the library of Memchached client culls the server which is working as a master. The Couchbase server stores, caches and replicate the data after arriving of referenced key.

(II) The administrator can request a redistribution of Virtual Bucket when machines are added in the cluster or removed from the cluster so that data are evenly spread across physical machines. [5]

The Couchbase server contain main object (A object that holds together to all column families) which is a subset of Virtual bucket that are used to distribute the data among the clusters. A mapping function [6] is used to calculate the Virtual Bucket, these mapping functions are also called a hash function that takes a key as input and prepare outputs for Virtual Bucket identifier. A table identifying the servers acting as Master and Replica servers for each virtual bucket. The number of Virtual Buckets in a Couchbase Server cluster must be more than the number of physical servers that present in the cluster. To add or remove servers configuration are needed because Couchbase server can fail at any time or any moment. To handle this situation the Couchbase Server cluster manager are used by running the Cluster management [7] code on every node in the cluster. The Couchbase Server cluster manager monitors health and coordinates data manager behavior on each node and it configures and supervises inter-node

behavior such as replication streams and rebalancing operations. Rebalance also was known as the systematic process of redistributing data within a live cluster. In Couchbase Server, the Rebalance Orchestrator rebalances by selecting and then migrating certain Virtual Buckets, including the data objects belonging to that Virtual Bucket, from old (Current) to new (Target) servers. Rebalancing will move both Master and Replica copies of objects. The intent is to spread the data and in particular I/O requests, evenly across the cluster. Rebalancing is typically done following the removal or addition of servers to a cluster. A Couchbase Server rebalances operation can be stopped and restarted any time. The Rebalance Orchestrator work within the Couchbase Server Cluster Manager which coordinates a Rebalancing process.

1.2 LOAD BALANCING IN COLUMN STORE DATABASES (CASSANDRA)

There are many types of Column Store Databases [2],
In Cassandra, the total amount of data managed by the cluster is represented as a ring [8]. The ring is divided into ranges equipollent to the number of nodes. The token range [9] is assigned by the cluster of nodes (which is essentially a range of hash functions which is defined by a Partitioner). The position of the ring and range of the data is determined by token [10] value. The data of the Column family is distributed among the nodes which are import on the row key. To locate the nodes with the token value the ring move along clockwise for determining the node where first replica of row will live. The token value is more powerful than the row key. To balance a load of servers with the avail of cluster manager by utilizing load balancing policies which is utilized to decide how to distribute requests among all possible coordinator nodes in the cluster. Particularly, they may fixate on querying "near" nodes (those in a local datacenter) or on querying nodes who transpire to be replicas for the requested data. The load balancing policies is contained by multiple hosts which forms as a Cassandra cluster which determine the driver will communicate with which hosts or coordinator node and with each incipient query which coordinator hosts or node will pick and which coordinator node or hosts utilize as failure. The node forms as a coordinator for the particular client operation when a node issues indites request after establishing a connection between client and node. A node becomes a coordinator host for the any client's operation when a node receives a client query. It provides the facility to communicate between all replica nodes responsible for the query and prepares and returns a result to the client. The Working methodology of the coordinator is to work as a proxy between the client application and nodes (or replicas) that own the data which has been requested. The ring should get the request on the cluster which is configured as a partitioner and replication strategy which is determined by the coordinator node. Let's deduced how this work. There is a unique key or primary key [11] engendered when utilizer update or engender data. The token range with the row key is generated by hashed. The data is then stored in the cluster n times (where n is defined by a number of machines in the cluster that will receive replicas of the same data). In Cassandra cluster there is multiple nodes work together. When a client send a request to the server a particular node receive this request and perform as a coordinator. To fulfill client requirement this coordinator may send this request to any node in the cluster then this coordinator become as a master and request receiver become as slave. This process is called master and slave. Every information i.e. health, status of each node has been updated to each node by using gossip protocol to find fault in any node.

I. Calculate the initial tokens predicated on the partitioner that you are utilizing. It can be manually engendered by equipollently dividing the token range for a given partitioner among the number of nodes.
II. These tokens are generated for Random Partitioner. Cassandra 1.2 and higher uses Murmur3Partitioner as default. These Murmur3Partitioner has a different key range.

1.3 LOAD BALANCING IN GRAPH DATABASES (NEO4J)

There are many types of Graph Databases [2],
Now-a-day many NoSql databases provide soi-disant sharding solutions [20]. Many people wants to use key-value or document based databases because these types of databases provide sharding functions solution that split-up the data across the many servers and these function store individual records. These types of databases do not fortify things such as referential integrity because they are not able to represent the connection between records with the help of query. These are reason that Graph databases such as Neo4j [13] are going to become the industry norm for the storage of any connected data. Thus, Neo Technology is working on an offering of a partitioned flavor of Neo4j subject to transmute.
In the Neo4j HA (high availability) architecture [16], there is load balancer [12] before cluster of neo4j database. In the cluster only one machine works as a master. The load balancer also known as a HA proxy determine route for all requests. When a machine fail which is elected as a master, an incipient server will be culled from the cluster as master and HA Proxy will automatically route transactions to this server. Load balancer known as HA Proxy will be configured with two open ports in which one for master to indite operations and another for operations. These two ports have different works with each application. When any application arrives, it perform indites with help of master port and perform reads with the help of slave port because each application two driver instances [13]. TCP of HA proxy has many feature that care of queries which take more to execute. These instances of database disconnected from the cluster when neo4j database inaccessible due to network or hardware failure and connected when neo4j database instances are available. If the master goes down another member are going to select and it have its role to switch from slave to master. The master will broadcast a message to all slave members that it is available and going to perform its job. Customarily an incipient master is elected and

commenced within seconds. During this time no indites can take place. Selection of master procedure is as follows

I- The slave which is highest committed ID will be selected as initial master if the previous master fails. This is a ensurity rule which through the slave becomes the initial master.

II- It is possible that after failure of master there are two slaves have the same priority to take place of master because they have the same highest committed transaction ID. In this situation the slave which has the lowest HA Server_ID will be selected as initial master. This is a good tie-breaker because of the HA Server_Id is unique within the cluster and sanctions for configuring which instances can become a master afore others.

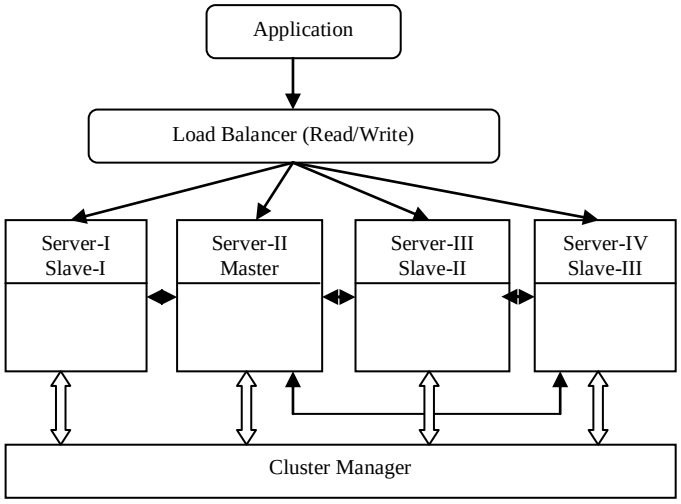

Fig. 1

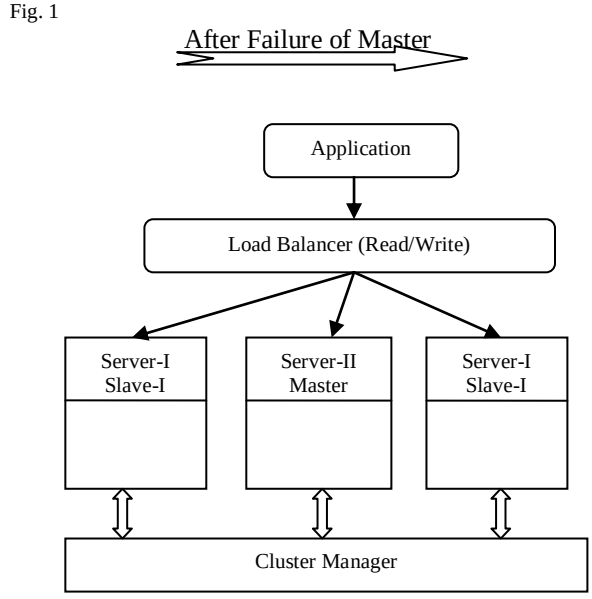

Fig. 2

1.4 LOAD BALANCING IN KEY-VALUE STORE DATABASES (VOLDEMORT)

There are many types of key-value store Database [2],
When the data fetch from disk of server, the data fetching time that is call seek time, may take more time if the more data are store in a particular disk. To remove this problem data are distributed on disks of the servers. When set of data are distributed into small pieces of chunk the cache efficiency will be increase. The server which holds the requested data must be routed because the recourse of the servers in cluster are not interchangeable due to this it may be possible that these severs become overloaded, hardware failure due to bad maintenance. If increase the data size and decrease the cluster size then the possibility will be increase of data loss. There are two ways for Load balancing will do their Job to get the stability on Load to balance the server we should do a measurement of two different things. One is the cluster health and the other one is the client health [14]. The VoldeMort database contain the simple load balancer which contains the different types of services such as “get client” [14] which work is to handle request of client if data is accessible for client then it will provide the accessibility if data is not available (due to overload, authentication, network failure etc) then it will throw this request to unavailable Exception service. One important thing should be to show here is that when “get client” get all URI (Uniform Resource Identifier) in the service’s cluster for each scheme, it will ask service’s load balancer strategy to load balance them. There are mainly two strategies used by Load balancer i.e. Random and Degrader [15]. The Random Load balancer use same machine for every request because it works under the development environment in which it chooses Random tracker (node) from the list of requested by the client if the list is empty it will return a null value. If we talk about another strategy of a Load balancer that degrader in which each node (client tracker) tracks many things like the number of calls, number of exceptions and percent of latency for given URI endpoint in the cluster to find out the health of node, whether to drop traffic if yes how much drop it. It depends on the average of latency per node of the cluster if the average is less than maximum cluster latency it will go through by calls. For example, if the threshold value mark of the load balancer is lower than the average of cluster latency value then we should drop the traffic data packets. The drop rate will decide that which node should be chosen to select the request i.e. if any node has lowest drop rate then it will be selected first. If the average of the latency per node is greater than maximum cluster latency then the balancer will allow to the node to drop the traffic. If we talk about cluster health monitoring, If the cluster latency exceeds high water mark[13] in a row we'll drop everything but there is some problems occur such as cluster may stick drop everything mode to prevent this problem there is a create mode available which through the traffic can pass . This is call cluster “Load Balancing Mode”.

References

[1] *Described in "Unix Review", March 1991, page 24, entitled "A 4GL Language".*

[2] *Retrieved from http://www.jamesserra.com/archive/2015/04/types-of-nosql-databases/*

[3] *Retrieved from https://dzone.com/articles/4-types-nosql-database*

[4] *Retrieve from http://www.couchbase.com/sites/default/files/uploads/all/whitepapers/Technical-Whitepaper-Couchbase-Server-vBuckets.pdf (VBUCKETS: THE CORE ENABLING MECHANISM FOR COUCHBASE SERVER DATA DISTRIBUTION)*

[5] *"Writing Views — Couchbase Server 3.0/3.1." Couchbase, docs.couchbase.com/admin/admin/Views/views-writing.html. Accessed 9 Jan. 2017.*

[6] *http://www.couchbase.com/sites/default/files/uploads/all/whitepapers/Couchbase-Server-Technical-Whitepaper.pdf (Couchbase Server Technical Overview)*

[7] *Ho, R. (2012, July 06). Everything You Need To Know About Couchbase Architecture - DZone Database. Retrieved January 09, 2017, from https://dzone.com/articles/couchbase-architecture-deep*

[8] *NoSQL Databases Defined and Explained | DataStax Academy: Free Cassandra Tutorials and Training. (n.d.). Retrieved January 09, 2017, from https://academy.datastax.com/planet-cassandra/what-is-nosql*

[9] *Maria Chalkiadaki and Kostas Magoutis "Managing Service Performance in the Cassandra Distributed Storage System" in IEEE International Conference on Cloud Computing Technology and Science 2013*

[10] *Elif Dede, Bedri Sendir, Pinar Kuzlu, Jessica Hartog, Madhusudhan Govindaraju "An Evaluation of Cassandra for Hadoop" from Grid and Cloud Computing Research Laboratory SUNY Binghamton, New York, USA in IEEE Sixth International Conference on Cloud Computing 2013*

[11] *Nishant Neeraj "Mastering Apache Cassandra" Published by Packet Publishing Ltd in October 2013 Birmingham, U.K*

[12] *Sanjay Sharma "Cassandra Design Patterns" Published by Packet Publishing Ltd in January 2014 Birmingham, U.K.*

[13] *Rik Van Bruggen "Learning Neo4j" Published by Packet Publishing Ltd in August 2014 Birmingham, U.K.*

[14] *Ian Robinson, Jim Webber and Emil Eifrem "Graph Databases" in O'Reilly Media Publication in june 2013, USA.*

[15] *L. (n.d.). Linkedin/rest.li. Retrieved January 09, 2017, from https://github.com/linkedin/rest.li/wiki/Dynamic-Discovery*

[16] *Design. (n.d.). Retrieved January 09, 2017, from http://www.project-voldemort.com/voldemort/design.html*

[17] *Eivind Siqveland Larsen and Knut Nygaard "Automatic scaling and maintenance of a NoSQL database" from Norwegian University of science and Technology in June 2014.*

[18] *Design. (n.d.). Retrieved January 09, 2017, from http://www.project-voldemort.com/voldemort/design.html*

[19] *https://en.wikipedia.org/wiki/**Keyspace**_(distributed_data_store)*

[20] *David Montag "Understanding Neo4j Scalability" in Neo Technology graphs are everywhere January 2013.*

Authors

Dildar Husain recieved his B.C.A. degree from Uttar Pradesh Rajshree Tondon Open University and M.C.A. from Maulana Azad National Urdu University Hyderabad. He is persuing M.Tech from Maulana Azad National Urdu University , Gachibowli, Hyderabad, India. His main research area is Load Balancing In NoSQL databases.

Mohd Omar received his B.Tech in Information Technology and M.Tech in Computer science from JNTUH. He is a Assistant Professor in School of Computer science and Information technology, Maulana Azad National urdu University,Gachibowli Hyderabad, India. His main research interests include software engineering, software testing, Image processing and distributed system.