Balanced Load in Distributed System with NoSQL Middleware

Samidha Joshi[1], Shubham Ameta[2], Geerija Lavania[3]
[1,2]Student, CSE, Jaipur Engineering College and Research Centre, Jaipur, Rajasthan, India
[3]Assistant Professor, CSE, Jaipur Engineering College and Research Centre, Jaipur, Rajasthan, India

Abstract : Due to the advancements in technology and the excessive use of smart phones, the data generated across the internet has increased. The explosion of data that we are witnessing currently has pushed the centralized data-management approaches to their limits and tested their durability. Pertaining to the current scenario, a change is required so as to scale the applications to levels that were unimaginable just a few years ago. But scaling alone isn't enough; companies also require all time availability of their applications and fast lightning, and this is where RDBMS databases fail. When load balancing is done in NoSQL databases, it becomes easier to store, manage and access the data from the data stores. This paper is focused towards the schemes to manage data in distributed databases using these technologies.

Keywords: NoSQL Databases, RDBMS, Load Balancing

I. Introduction

Relational Database Management System (RDBMS) was once the predominant technology for maintaining structured data in various applications. These data stores have been relying on relational calculus. The facilities that they have been providing by SQL were widely adopted. At that time they were thought as the only alternative for data storage which is to be accessed by multiple clients consistently. [13]
In the last few years, there has been an immense rise in the generation of data sets due to escalation in the mobile devices using internet and advancement in the web technology. These data sets need to be stored so that they can be accessed when required. [6]

The conventional RDBMS hasn't proved to be the most appropriate solution to manage the oversized massive amount of data generated by various businesses. If we talk about large data, the term "Big Data" comes into the picture. Big Data is a phrase which is used to refer to extremely large amount of complex data. It includes all the variety of data structures like structured, unstructured and semi-structured data. [6] NoSQL is used when we are working with a large amount of data and the data need not be of relational model type. Major companies like Amazon, Facebook etc. could not cope up with the solutions provided by RDBMS. In order to overcome this problem the need for the development of NoSQL arouses. [19] However the need to handle large amount of data has arisen in a few years. This is where NoSQL databases came into use. This term was introduced in 1988 for a database which was not a relational database and stands for "Not Only SQL". Such a type of database was based on a BASE Model that is, it is **B**asically **A**vailable, has a **S**oft state and is **E**ventually consistent. [5] It was proposed by Brewer which trades off the rigid properties of consistency and isolation in order to support availability, speed and agility. Load balancing helps in accessing more than one server. If designed correctly, it becomes a high availability solution. Redundancy will be the least in case of spreading of load in a database. Therefore, in case of a server breakdown, only the users on the broken server will lose the connection and the rest of the users will be able to continue the work. But access to services is "critical". So load balancing along with the solution of redundancy has proved to be more efficient in the IT industry. [9]

II. Literature Review

In 1965, Multi Value databases were introduced at TRW. Then AT&T released DBM in 1979. The term NoSQL was initially introduced in 1998 to name the open-source database. The databases which did not use the sequential query language (SQL) deliberately were seen as a deviation from the traditional databases, as Carlo Strozzi[17] explains in his article. [13] A graph database was started in 2000, named Neo4j while Google's BigTable started in 2004. MongoDB was introduced as a part of an open source cloud computing stack and as the first standalone release in 2009. Cassandra project was started by Facebook in 2008 as well as the Project Voldemort. [1]

This term NoSQL re-appeared in 2009 when Eric Evans used it to describe the sudden rush towards the non-RDBMSs. [7] He is often regarded as the person who made this term famous and believed that NoSQL is apt for those situations where the relational databases do not fit. [13]

In another research article, the author has specified that NoSQL technology is gaining importance because of Cloud computing, the Web, the Big Data and the Big users. According to a statistical report in 2014, around 3 billion people spent 35 billion hours per month on the internet and with the onset of several mobile and web applications, this rate is gradually increasing. This practice creates a large amount of unstructured data. NoSQL provides facility of processing and analyzing data. [16] RDBMS scales up the data by addition of hardware and processing equipment and at a point, it stops scaling. NoSQL scales down the data by two methods- partitioning and replication.

Partitioning is the method in which the data is divided into smaller parts so that speed can be increased and performance can be improvised, whereas replication is the process of creating similar data sets to be stored at different locations which enhance accessibility and fault tolerance..Load balancing is the distribution across the world. In order to obtain maximum throughput from the network, the load has to be distributed and tasks have to be partitioned among the various nodes connected. [10]

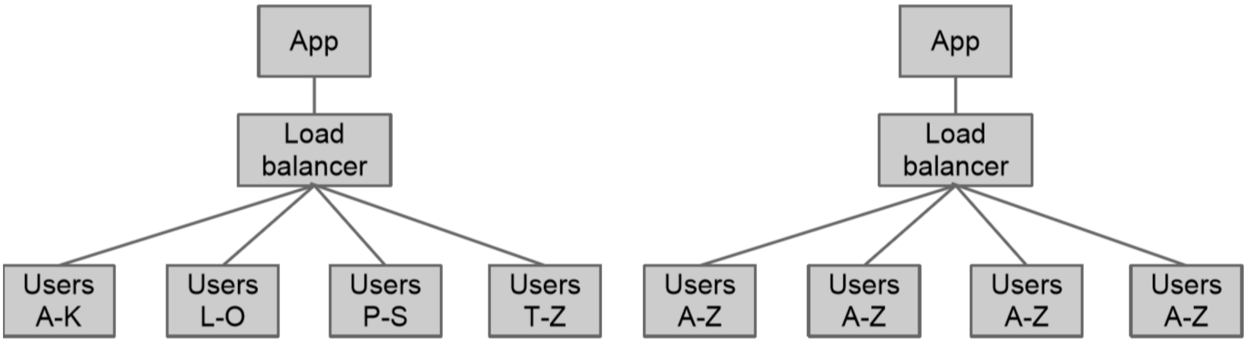

Figure 1: Partition and Replication in NoSQL Databases [1]

It is clear that with several nodes, work is bound to get unbalanced. Some nodes will receive data all the time, while other may remain idle. Thus, to ensure that the processing elements are neither overloaded nor they remain idle, the load balancing algorithms are required. They migrate the excess load from the processor to another processor which has load below the threshold load. [10]

Now, to determine which node is heavily loaded and which one is lightly loaded, load estimation is required. This can be estimated by measuring processing power of the nodes. Processing power not only means the processing speed of processor but also the node's overall configuration. Different load balancing algorithms have different complexities, which can be measured by the amount of communication done to locate the least loaded node. In general, load balancing algorithms prevent any node from being idle. [12] The quality of these algorithms depends upon two factors –

i) the number of steps required to achieve the balanced state.

ii) The amount of load that is to be moved across the nodes. [10]

The more information it collects, better decision will it make. Load balancing has following advantages-

- It enhances the performance of each node.
- Starvation will not occur for small tasks.
- Mean job time is reduced under job transfer overload.

It is done in two situations – Firstly, when the distribution of tasks is done before the system goes into runtime. It is called static load balancing (SLB) algorithm. Communication over-heads is negligible as there is no reassignment here. [10] They collect no information and make balancing decisions based on probability. [12]

Secondly, during the execution time, dynamic load balancing (DLB) algorithm occurs. [10] These algorithms collect varying state information to make their decisions. [12] Communication over-heads occur when redistribution of the tasks occur at the same time as the distribution of load from nodes with heavy data to the ones with the light data. [10]

Static Load Balancing

In such an algorithm, the processes are assigned to the processors during the compile time based on the performance of the nodes. The assignment of processes is fixed. An algorithm of this type has fixed number of jobs and does not need information about the nodes. A number of factors are considered before any task is assigned to the processors – incoming time, resource needed, mean execution time and inter-process communications. Measurement of these factors is done during the compilation time. Therefore, static load balancing algorithm is also called probabilistic algorithm.

One of the advantages of this algorithm is that there are no or very less overheads created at run time as there are no changes of tasks later. It has been observed that load balancing will be better if the number of tasks is higher than the number of processors. [10] Following Figure shows static load balancing where the tasks arrive at the assignment queue.

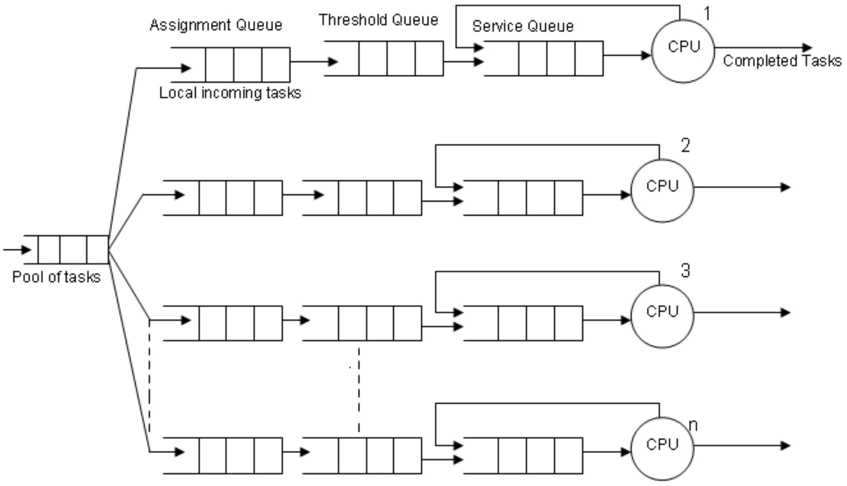

Figure 2: Model of Processing Node [source: C. LU and S. LAU, 1996.] [10]

Dynamic Load Balancing

One of the major disadvantages of static load balancing is that it requires excess information about the system, processors and tasks, which may or may not be available at the time of compilation. An in-depth study of the system needs time & is tedious in nature. Hence, dynamic load balancing came into being where tasks were assigned to machines at run time. Evidently more overheads occur in this case and it increases as the number of processors increase. [10] The algorithm collects information about system state and the task. Only after the process execution, will it make a decision. This decision will be better if it can collect more information about the machines in shorter period of time. [12]

Load Balancing in NoSQL Data Stores using DBalancer

Load balancing is done when there is no guarantee of whether workload will be constant or dynamic. DBalancer is a distributed module, which can be installed on the top of a typical NoSQL data store and can provide an efficient load balancing mechanism. This is done by simple exchange of messages and data movement operations. Various skewed and dynamic workloads can be applied on NoSQL clusters to implement the load balancing algorithms. NoSQL systems provide load balancing by re-partitioning and re-distributing the data items. [4] DBalancer has the following useful-features:

Datastore Abstraction: It does not depend on the underlying NoSQL data stores. The users need to define set of specific actions, such as movement of item and routing table commands, so as to take the advantage of the DBalancer features.

Algorithm Abstraction: To define new load balancing algorithms, a set of messages and actions along with the suitable trigger conditions is given.

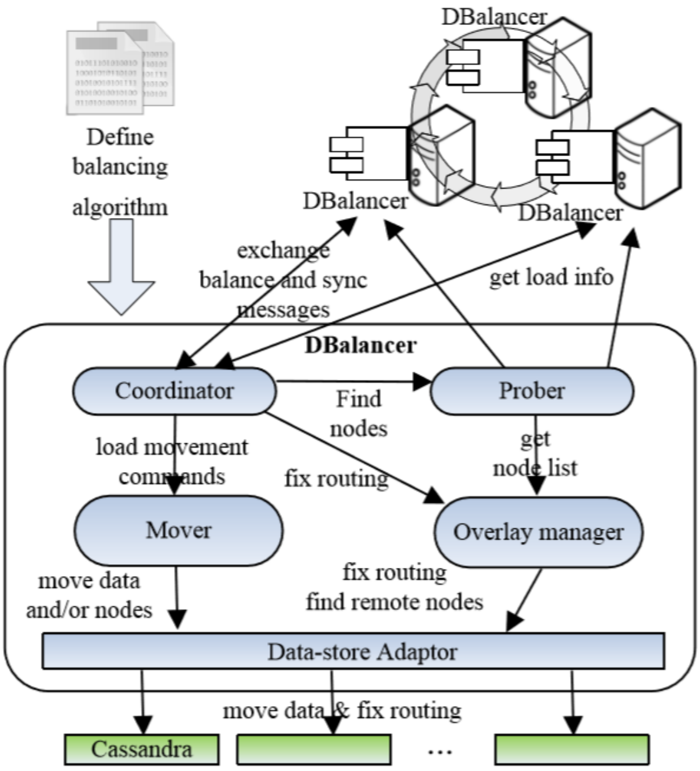

Figure3: TheDBalancer Architecture [4]

Load Balancing Approaches

Amongst the varying techniques for load balancing the two major methods have been discussed to balance the load in the distributed systems namely – Load Balancing Query Processing (LBQP) approach and Resource Allocation approach.

LBQP Approach

This approach is categorized in two phases:

1. Static planning that is done at compile time
2. Dynamic allocation of processes done at run time. [11]

The first phase called the static planning phase consists of a relational query which acts as an input for this phase. The output of this phase known as *logical processing plan* is a sequence of relational operations. The second phase called the dynamic allocation phase is executed at run time. The logical plan serves as an input here and the algorithm selects a physical location for each relation in the plan. The output produced at this phase is known as *global processing plan*. The third and final phase of LBQP is the refining phase. For each join in the global plan which has relations at two different locations, this phase chooses between join-based and semi join-based methods. Easy and simplified query optimization is one of the major merits of LBQP approach. [11]

Resource Allocation Approach

As S. Jagannatha, D.E. Geetha& Suresh Kumar proposed in their methodology, load balancing can be analyzed by considering sharing of resources, allocation of fragment replicas and transaction in distributed system. [15] The two major database design phases are: fragmentation and allocation. Fragments are the small non-overlapping portion of global schema. They are allocated to the nodes, where needed using horizontal, vertical and mixed fragmentation. The fragmentation replica improves performance and availability by reducing waiting time and by concurrent processing. The replication control methods are intended to increase average response time without compromising on data consistency. [15]

Light-weight Design: The DBalancer is designed to have the least effect on any node under processing. To coordinate the load balancing procedure, only a number of messages are utilized.

The DBalancer component is executed in the node of every store and it is configured with the desired load balancing algorithm. Messages are exchanged to find partners and load balancing procedure is performed. (As seen in upper part of figure) After nodes are found and reserved, keys are exchanged. (As seen in lower part of figure)

III. MARKET EVALUATION

F5 mentions in its paper that without databases businesses would grind to a halt. But the structured data represents overall information about an organization, yet here are more efforts to be taken for effective load balancing for mainstream DBMSs. This is because of transactional and critical nature of databases. There have been advances in database technology and the services of Application Delivery Controllers (ADCs) combined have proved to be a major boost for the marketplace. As ADCs offered load balancing and other functionalities such as monitoring, security etc. combined with the growth in database clustering and high availability at database levels, this led to ADCs being used for load balancing of DBMS products. But there are some challenges to load balancing DBMSs. Firstly, DBMS can access all the records of a particular table and thus, it is updated directly, all at once. Therefore, it is an issue to perform load balancing such that all instances have access to all data for all tables. DBMS also has transactional reliability, which guarantees relevance of all the transactions which are complete. Load Traffic Manager (LTM) is an ADC of F5 uses many monitors to check the clusters. If the primary and secondary clusters are configured as members of a single pool, one will go down and the other will receive the traffic automatically. It provides the connection address for applications utilizing the database. [8]

In its white paper, **DataStax** provides the general implementation strategy for NoSQL in various enterprises. It describes why enterprises are using these NoSQL solutions and what are the problems faced by them in implementing these solutions. It is a known fact that improved response time for external systems directly affects the customer satisfaction and revenue. Apache Cassandra, a hallmark of NoSQL solutions, has the ability to interpret and write data much faster than any RDBMS. Also it can deliver query results just as fast across data on a large scale. For the same reason, eBay uses DataStax Enterprise, which is powered by Apache Cassandra. EBay has also replaced the traditional RDBMS with DataStax Enterprises and now it meets the response time by managing 250 TB of data and servicing 6 billion writes and 5 billion reads per day. [3]

Nomadix published the paper and described the way to use Nomadix Service Engine (NSE) for load balancing. Load balancing is the process of assigning the guests, any one of the multiple connections. This in hotel management is done to provide connection to users. It is up to the hotel, which guests will use which type of connection, and also what the users are allowed to do with that connection. To aggregate five low-speed links to create one high speed link with high quality NSE configured with load balancing between all the links. [9]

As **CouchBase** explains in its paper as to why use NoSQL, it states that companies and enterprises are using NoSQL technology as it provides data management facilities - Improved development of application with a more flexible data model; Greater capability to dynamically scale the data; and Better performance for the user satisfaction who want applications with high response and more complex data processing. [2]

IV.CONCLUSION

It is never meant that the NoSQL databases are replacing the relational databases but it is a fact that the former are designed to handle the large amount of data that cannot be handled by the latter. There are various other technologies that are used to handle complex data – one of them is using load balancing in Distributed databases. Balancing large and dynamically changing numbers of user requests in distributed systems has always been the area of focus.

Unexpected load spikes or skewed data access patterns led to the need of Load Balancing, and thus resulting in severe degradation in performance of data serving applications. But NoSQL has gained popularity as it is one of the rising techniques in this field. It needs many improvements though as it has its own share of disadvantages. Few of them being narrow focus i.e. it mainly performs storage but very less functionality; NoSQL databases requires much more complex Data management as compared to relational databases; Backup is not easy as there is no recovery mechanism in databases like MongoDB.

In this paper, various approaches of implementing load balancing have been discussed. Most importantly, the methods when applied using NoSQL technology, have been emphasized. The fragment allocation and resource allocation strategy are considered to minimize the cost of computation and for uniform workload distribution. There is much more scope in terms of data managing techniques and databases can be handled in a better way. These technologies have contributed in a great way in this field.

V. REFERENCES

[1] ArtoSalminen Introduction to NoSQLNoSQL Seminar 2012
[2] Couch Base, “Why NoSQL”, whitepaper
[3] Implementing a NoSQL Strategy, White Paper by DataStax Corporation, July 2013

[4] IoannisKonstantinou, DimitriosTsoumakos, IoannisMytilinis, NectariosKoziris, "DBalancer: Distributed Load Balancing for NoSQL Data-stores", SIGMOD'13, June 22–27, 2013, New York, New York, USA. Copyright 2013 ACM 978-1-4503-2037-5/13/06.

[5] João Ricardo Lourenço, Bruno Cabral, Paulo Carreiro, Marco Veira and Jorge Bernardino, Choosing the right NoSQL database for the job: a quality attribute evaluation Published on 14th August 2015.

[6] Katarina Grolinger, abhinavtiwari, Wilson A. Higashino, Miriam AM Capiretz "Data Management in Cloud Environments: NoSQL&NewSQL data stores", Journal of Cloud Computing, a SpringerOpen Journal 2013

[7] Knut Haugen A brief history of NoSQL All about the Code, 16th March 2010 http://blog.knuthaugen.no/2010/03/a-brief-history-of-nosql.html(last seen: 20th February, 2016)

[8] Load Balancing Oracle Database Traffic, Updated April 10th, 2012 by F5

[9] Load balancing White Paper, Nomadix, Options for handling multiple ISP Lines at hotels.

[10] Md. Firoj Ali and RafiqulZaman Khan. "The Study On Load Balancing Strategies In Distributed Computing System". International SJournal of Computer Science & Engineering Survey (IJCSES) Vol.3, No.2, April 2012.

[11] Michael J. Carey, Hongjun Lu, Load Balancing in a Locally Distributed Database System, Computer Science Technical Report #625, December 1985.

[12] Parveen Jain, Daya Gupta, An Algorithm for Dynamic Load Balancing in Distributed Systems with Multiple Supporting Nodes by Exploiting the Interrupt Service, International Journal of Recent Trends in Engineering, Vol 1, No. 1, May 2009

[13] Prof. Walter Kriha, "NoSQL Databases" Chapter 1-2

[14] RenuKanwar, PrakritiTrivedi, Kuldeep Singh, "NoSQL, a Solution for Distributed Database Management System", International Journal of Computer Applications (0975 – 8887) Volume 67– No.2, April 2013

[15] S. Jagannatha, D.E. Geetha, T.V. Suresh Kumar, K. RajaniKanth; Load Balancing in Distributed Databases System using Resource Allocation Approach International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 7, July 2013

[16] SQL vsNoSQL- 4 Reasons Why NoSQL is better for Big Data applications, 19 Mar 2015.

[17] Strozzi, Carlo: NoSQL – A relational database management system. 2007–2010. – http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/nosql/Home%20Page(last seen: 20th February, 2016)

[18] Whitepaper Giritech , " Load Balancing and Redundancy Version 5".

[19] Whitepaper HP "There is no free lunch with distributed data" retrieved on date December 2018.