International Journal of Engineering & Technology, 7 (4.19) (2018) 223-225

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Load Balancing in Cloud Systems Using Dynamic Chained Failover Algorithm

[1*]M.Sangeetha, [2]K.Senthil Kumar, [3]G.Parimala

[1,2,3]Assistant Professor, Department of Information Technology, SRM University
**Corresponding Author E-mail: [1]sangeeta.mani@yahoo.com*

Abstract

Large-scaled cloud systems have been there in various domains by extendinglarge number of resources. Efficient allocation of shared assets in cloud is a vital but provocative issue. It is known that existing load balancing policies like Random,size-based polices, Join Shortest Queue are implemented in cloud as they are simple and efficient. The performance of the above mentioned policies decrease when workloads are temporally correlated. We propose the new load balancing method, Dynamic Chained Failover Algorithm in this paper where particular time period at which the server is getting overloaded is taken and particular task which causes overloading in particular interval is placed in all servers as a replica instead of data reallocation. Overall system performance is increased. Exploiting capable server to improve the system performance is thus demanded.

Keywords:*Load Balancing, Cloud System, Servers, Overloading.*

1. Introduction

In the computer system, a cloud is a large storage database where there are number of servers. Hence, the users can save their datasecurely. Load balancing is needed to balance the load between all servers. Load balancing achieves the balancing of the servers by moving the processes or tasks to lightly loaded servers from overloaded servers so that it increases the performance of the system. In DCF Algorithm, we are achieving load balancing the server which is overloaded, it checks in nearby server for replica of each of its blocks to reassign the task. The performance of DCF algorithm is more efficient and works smarter than the existing algorithms.

2. Related Work

2.1. Scalable Load Balancing in Cloud Storage Systems

Centers of Cloud System consists of thousands of nodes/servers which provides large bytes of storage to numerous users and domains. In this type of systems, there are two difficulties-(I) objects which are stored becomes dynamically popular(II) the cost for reconfiguration is high because of subscription of bandwidth [1]. The storage systems that are existing are howevernot suitable to address these difficulties as there are numerous servers and objects in cloud. Ursa is a method whichcome up with the selection of sub objects from the serversand also helps in reducing the costs for reconfiguration.
This method explains the way of designing, implementing, evaluating Ursa. This helps in scaling up to a huge number of storage servers and objects and helps in minimizing latency and also decrease cost for bandwidth. As appropriate optimization is high in cost, they organize scalable techniques for approximation in selecting nodes and objects [1]. This evaluation proves Ursa helps in saving cost-while load balancing and scaling to number of large storage systems and also time effective and plays efficient role in making decisions

2.2. Rebalancing of Load in Distributed File Cloud Systems

The critical part of building blocks of distributed file systems in cloud computing is based on map reduce paradigm algorithm [2]. In this type of filesystems, the concurrent nodes simultaneously thathelp in cloudcalculating a file is divided into a number of lumps that are allocated in different nodes in order to perform the jobs of Map Reduce parallel. But, in theenvironment of cloud computing, serverscan be upgraded or replaced or added in the clod system. In this method, files can be created dynamically. It can be deleted and concatenated. This achieves misbalancing in load in a distributed file system i.e., the file lumps are not shareduniformly among the servers. In production, this type of emerging file systemsdepends on a central server for reallocating lumps. This type of dependency is poor in a large-scale system as the central load balancer is kept under appreciable workload that is scaled linearly with the size of system. Soit becomes the performance bottleneck and the failure at a single point. In this method, the problem of load unbalancing is coped up completely [2]. This algorithm is compared with approach of centralized system in production and a contending distributed solution submitted in the literature. Thismodel results in achieving our proposal that is comparable with the existing approach and substantially performs the earlier distributed algorithm in terms of load imbalance, cost. The performance of the proposed algorithm is implemented in the Hadoop file system and analyzed in a cloud environment further

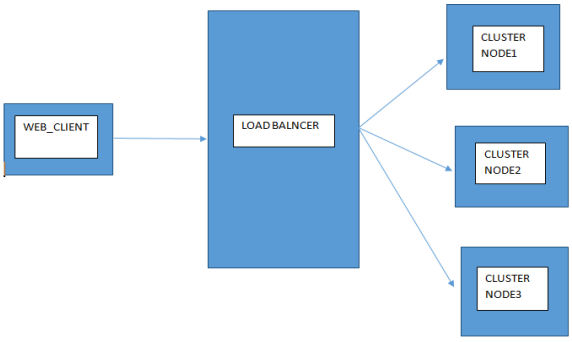
2.3. Data Processing on Large Clouds in Simplified manner

Map Reduce is a scheduling model is used for treating and creatinghuge data sets. Users can specify a map function that treats a keyvalue match to createan intermediate key value set, and a decrease function that unites all intermediate values under the same intermediate key value [3]. Most of the real world processes are describable in this method, as explained in the paper. Programs done in this type of functional way are executed on a large cloudsystem in parallel way. The run-time machines look about the details of dividing the input data and scheduling the execution across different machines. It also manages failures in systems, andintermediate communication between machines This helps programmers in using the resources in large distributes cloud system without any experience [3]. ThisMap Reduce implementation runs on a cloud system that is hugeand achieves scalability. This is a distinctive Map Reducecalculation that helps in processing large data on multiple machines. This system is very easy to use such thatthousands of Map Reduce programs have been enforced and more than one thousand Map Reduce jobs are performedon Google per day to day.

3. Proposed System

In this proposed system, heavily loaded websites must serve lot of requests, if not more than millions of simultaneous requests from users and deliver the exact data such as text, images or video, etc., all in efficient manner. To achieve this effectively, it requires additional servers to be added in modern computing systems. The load balancer behaves as the "traffic cop" watching your server and it routes the users requests all over servers which are capable of accomplishing these requests so that it increases speed, performance and capacity utilization and assures that none of the server is overloaded, which could decrease performance. Even one server goes down, the load balancer redirects the requests to the left over servers. When a new server is imparted to the server list, the load balancer mechanically starts to send the traffic to the server. In this way, a load balancer executes the user requests or network load reliably across all the servers and ensures high availability by sending requests only to servers that are available. When there is a case that all the servers in the cloud are overloaded, it provides the tractability to add servers as needed and achieves high reliability, and the load balancer will ensure that the requests will be directed to the newly added servers.

4. System Architecture



5. Dynamic Chained Failover Method

5.1. Cloud ServiceProvider (CSP)

The CSP controls a large are small scale of Servers. In this Project We cannot provide Authenticated Server for services for that purpose we use MySQL Database As server and analytically implemented our business logic in java coding Technique's. In the First Step the Up-loaders they are just User's according to CSP, They need to register for a service provided by the cloud when they Register they Provide a Dummy amount and choose the Server Space. Once Registration over, the CSP has two types of authentications. They Are-Activate (Allow) and Not-Activate (Not-Allow). Allow: When the buyer register he will be given permission by the provider, the provider will check for the details. The Main Reason For this is to Control Unwanted users or Unknown user. The CSP can able to check the details about the Up-loaders. Not-Allow: If the Up-loaders is not allowed. He/she can't able to login to their home page. Validation occurs. Any time the CSP can De-activate the Accounts. If the up-loaders find suspicious.

5.2. File -Up loaders

The Up loaders is the person who should be registered to access the server (OR buy a server with some cost). If the **Up-loaders** is activated by the CSP they can use the server. Depends upon the space he bought. The buyer® user will uploads Files (R something). Each time the Uploaded Content will be checked using Business logic ie the space of the file will be checked and store to the server (if space available). If the file is higher than the bought server space. The user cannot upload the File. (Server Error occurs) The server will escalate the files giving error.

i) Servers (Database)

Servers are in a distributed environment, For the Project we use Database as a Server. The Server can be added by the CSP Each Server has same amount of Space Availability.

ii) Flow

This Modules shows the CALV process. Suppose there are 2 servers A&B in space 10mb, A uploader uploads a file which is in space 2mb it will get upload in first server A. Now the Server B has large space 10 for(B) & 8 for(A), The next user uploads another File which is 1 mb. Now the Algorithm checks whether A is Greater than B it's false now,

Again next user uploads the files if the condition satisfies A will come to first preference to Store.

It is observed that in peak time the servers may be overload. Particular server and task which causes overloading is identified and load balancing is done by below algorithms-

Particular time period at which the server is getting overloaded is taken.

- Particular task which causes overloading in particular interval is placed in all servers as a replica instead of data reallocation.
- At particular time, the load balancer gives more overloaded servers (measured by sum of(Lek-Ax). The value higher is more overloaded.
- At interval, Calculate all epoch in all the servers. If the value is more than and closest to, (Lek-Ax) it needs to be reassigned
- If no values are closer or higher than this, just arrange all the tasks in descending order of their values and reassign tasks from top to down.

Some intervals may not be predicted. Some tasks in next interval may not be identified and overloading may occur. In order to prevent this, below algorithm is used.

- When the server become overloaded, it checks in nearby server for replica of each of its blocks to reassign the task.
- If there is no replica found, it creates a data block in the server which has the computing capacity. It checks whether the computing capacity is available based on formula (Ax-Lek)>= Clime, for each epoch ek belongs to T.
- If server sj receives positive responses from several servers, it redirects task ti to the server with min value (Ax-Lek)- Cti;ek
- In this way, the selected server has sufficient computing capacity to host task ti while its resources are more fully utilized.

A.

If server si becomes non-overloaded after the task redirection, the DLB process stops.

- Otherwise, the next task is picked and the same process is conducted.
- If all the servers are filled, then a new node is created to store data.

6. Performance of Dynamic Chained Failover Method

The performance of Dynamic Chained Failover Algorithm is as below. The performance doesn't depend on the type of task and job scheduler; we test the performance of DCF with only FIFO scheduler.

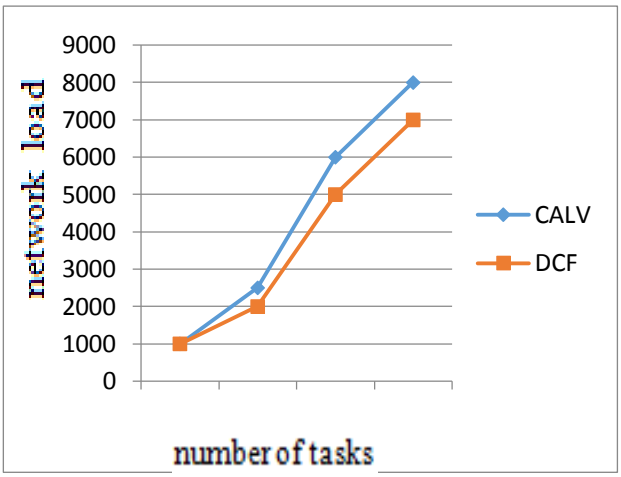


Fig. 1: (a)

The above graph shows the network load during number of tasks. Y-axis represents network load .X-axis represents number of tasks. Due to unexpected tasks, several servers may be overloadedin CALV (existing algorithm).Hence, more tasks need data transfers in running time in CALV, leading to higher network load. But in DCF,there is no possibility of servers to be overloaded as the tasks are shared among the servers dynamically. It means that DCF can effectively handle both servers caused by unexpected tasks. Figure 1(a) shows the network load for data transfer during job running versus the number of jobs.

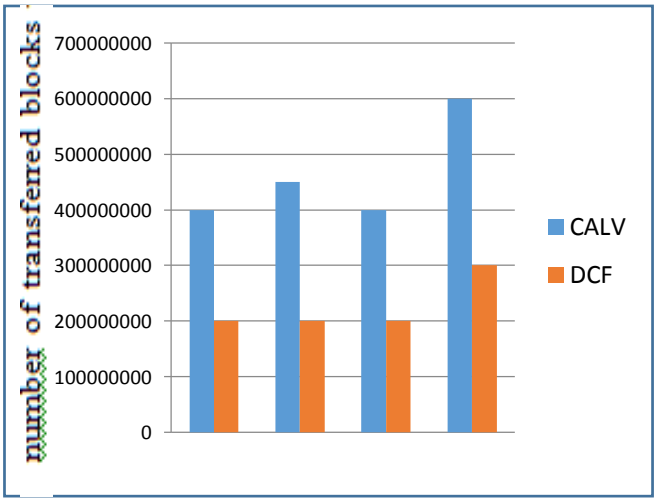


Fig. 1: (b)

Figure 1(b) shows that the number of transferred blocks versus the number of jobs. It indicates that CALV(existing algorithm) has a slightly higher number of transferred blocks than DCF (proposed algorithm). This is because in CALV, a server may need to replicate a data block to another server in order to redirect a task to the server, whichGenerates block transfers whereas DCF generate much fewer block transfers because it reassigns tasks to their other data servers without moving data blocks.

7. Conclusion

This paper describes about the load balancing technique in cloud environment efficiently. This method involves in identifying the server which becomes overloaded in peak time and helps in overcoming the overloading problem by balancing the load between the servers using dynamic chained failover algorithm. It moves the tasks efficiently from high loaded to low loaded servers. This method has more advantages and it works efficiently.

References

[1] G. You, S. Hwang, and N. Jain. "Scalable Load Balancing in Cluster Storage Systems". In Proc. of Middleware, 2011
[2] H. Hsiao, H. Su, H. Shen, and Y. Chao. Load "Rebalancing for Distributed File Systems in Clouds". TPDS, 2013.
[3] J. Dean and S. Ghemawat. "Map reduce: Simplified data processing on large clusters". 2004
[4] Abad, Y. Lu, and R. Campbell. Dare: "Adaptive data replication for efficient cluster scheduling". In Proc. of ICCC, 2011.
[5] Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng. Cdrm: "A cost-effective dynamic replication management scheme for cloud storage cluster". In Proc. of CLUSTER, 2010.
[6] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz. "The Case for Evaluating MapReduce Performance Using Workload Suites". In Proc. of MASCOTS, 2011.
[7] Z. Li and H. Shen. "Designing a hybrid scale-up/out hadoop architecture based on performance measurements for high application performance". In Proc. of ICPP, 2015.
[8] Y. Chen, S. Alspaugh, and R. Katz. "Interactive analytical processing in big data systems: A cross-industry study of map reduce workloads." Proc. of VLDB, 2012.