An Approach towards Secure Load Balancing in Cloud Computing

V.T. Lanjewar[1], Prof. R.V. Dharaskar[2]

[1] ME Student, PG. Dept. of CSE, SGBAU, Amravati, India.
Email-id: vrushali.lanjewar@gmail.com
[2] Associate Prof., PG. Dept. of CSE, SGBAU, Amravati, India.
[2]Email-id: rvdharaskar@rediffmail.com

Abstract: **Cloud balancing is still new, but the technology to add value is available today. The main objective is to develop an effective load balancing algorithm using task scheduling to maximize or minimize different performance parameters such as throughput, latency for the clouds of different sizes. This work has also been enhanced with divisible load balance scheduling which has resulted into efficient load balancing with minimum execution time. In order to authorize the data hosting by genuine user, here in this work a noble public key cryptography scheme has been employed. RSA algorithm has been employed in this work for ensuring authentication of users to store data on public cloud.**

Keywords: Authentication, Hadoop, Load Balancing, MapReduce, Kerberos

I. Introduction

Cloud computing is a term, which involves virtualization, distributed computing, networking, and software and web services. A cloud consists of several elements such as clients, data center and distributed servers. It includes fault tolerance, high availability, scalability, flexibility, reduced overhead for users, reduced cost of ownership, on demand services etc. Central to these issues lies the establishment of an effective load balancing algorithm. The load can be CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. This technique can be sender initiated, receiver initiated or symmetric type (combination of sender initiated and receiver initiated types). Hadoop is one of the platforms for big data.

Hadoop includes two parts, namely, the Hadoop Distributed File System (HDFS) and MapReduce. Data is stored in HDFS as blocks that are of fixed sized by default. There are two kinds of nodes in Hadoop, where a namenode serves as the master node and datanodes as slave nodes. The data blocks are located on datanodes and Hadoop partitions the datanodes on different racks. Hadoop installed on a very large cluster can span many datacenters including many racks. The MapReduce framework has proven to be a powerful and cost-efficient strategy for massively parallel data processing.

Fundamentally, a MapReduce job is executed through two primary phases. In the *map* phase, a function is applied in parallel to data from various input datasets. This function yields intermediate results in the form of a list of key-value pairs. Pairs with the same key are subsequently grouped together and allocated to a reduce task based on a *partition function*. In the *reduce* phase, the reduce task runs in parallel over each key group to produce the final result

Challenges

One problem with Hadoop is balancing the workload. In Hadoop, the uneven distribution of data to the datanodes results in load imbalance between different racks resulting in reduced performance. There are multiple challenges involved in the implementation of a fully functional cloud balancing strategy. Some of these challenges are a result of the immaturity of current cloud-based offerings, and, as such, they might be automatically addressed as cloud environments continue to mature based on market demand and experience. Other challenges, however, are likely to require standards before they will be sufficiently addressed.

II. BACKGROUND

The evaluation examines the benefits and trade offs associated with each policy. Less aggressive policies are able to provide zero overhead rebalancing at the expense of leaving the deployment in a non desired state for longer periods of time[1]. *Liu* shows the improved algorithm can really balance the overload racks much quicker than the Hadoop algorithm and the improved algorithm can also distribute the data more evenly to each machine[2]. *Wang* focuses in particular on algorithms based on closed queueing networks for multi-class workloads, which can be used to describe application with service level agreements differentiated across users [3]. Security analysis shows that both *AnonyControl* and *AnonyControl-F* are secure under the decisional bilinear Diffie–Hellman assumption, and the performance evaluation exhibits the feasibility of the schemes. In experimental scenario with the number of task requests increased [4]. There is need of an algorithm which can offer maximum resource utilization, maximum throughput, minimum response time, dynamic resource scheduling with scalability and reliability. This work proposes an autonomous agent based load balancing algorithm (A2LB) to address above issues.[5] *Zhao* has verified the load balancing effect of DLB and LB-BC by comparing their percentages of the incremental standard deviation values. These received requests constitute the total workload to be handled by the cloud data center and aren't the load of some physical host [6].

This paper introduced The rest of this paper are organized as follows: **Introduction** gives an approach towards achieving secure load balancing in cloud S**ection I**. **Background** gives a premise of the proposed research problem is pointed out, and then the proposed problem is formalized in detail in **Section II**. **Previous Work done** gives related work of the current approaches achieving load balancing of cloud data centers will be introduced briefly in **Section III**. **Section IV**. **Existing methodology** discussed in **Section V** Analysis and Discussion discuss In **Section VI, Proposed methodology** gives the design and implementation process of the proposed algorithm in detail and **Possible Result** in discussed. Finally, **Conclusion** of the paper discussed in **Section VII**.

III. PREVIOUS WORK DONE

Most previous works, generally, utilize a series of algorithms through optimizing the candidate target hosts within an algorithm cycle and then picking out the optimal target hosts to achieve the immediate load balancing effect. However, the immediate effect doesn't guarantee high execution efficiency for the next task although it has abilities in achieving high resource utilization. *Duplyakin et al* (2013)[1], presents an environment that manages multicloud deployment rebalancing by terminating instances,in lower preferred clouds and launching replacement instances in higher-preferred clouds to satisfy user preferences. *Liu et al* (2013)[2] , focused on the overload machines and propose an improved algorithm for balancing the overload racks preferentially. The improved method constructs Prior Balance List list which includes overload machines, For BalanceList list and NextForBalanceList list by many factors and balances among the racks selected from these lists firstly. *Wang et al* (2014)[3], experimentally explores the relation between probabilistic routing and weighted round robin load balancing policies. From the experiment a similar behaviour is found between these two policies, which makes it possible to assign the weights according to the routing probability estimated from queueing theoretic heuristic and optimization algorithms studied in the literature. *Jung et al* (2015)[4] proposes a semi-anonymous attribute-based privilege control scheme *AnonyControl* and a fully-anonymous attribute-based privilege control scheme *AnonyControl-F* to address the user privacy problem in a cloud storage server. *Singh et al* (2015)[5] work focuses on load balancing in cloud computing environment. Load balancing in cloud computing has been ignored, but rapid growth in number of cloud users has raised demand for load balancing mechanisms. This work has proposed an autonomous agent based load balancing mechanism which provides dynamic load balancing for cloud environment. *Zhao et al* (2016)[5] focused on the selection problem of physical hosts for deploying requested tasks and proposed a novel heuristic approach. The Bayes theorem is combined with the clustering process to obtain the optimal clustering set of physical hosts finally.

IV. EXISTING METHODOLOGY

Multicloud architecture: The architecture that designed consists of four main components: (1) a workload management system (including a job scheduler and workers), (2) sensors to monitor demand, (3) policies to scale the number of deployed instances up or down, and (4) an auto scaling service to enforce the chosen policy. An auto-scaling service, Phantom is use to launch and monitor instances acrossmultiple clouds. [1]

Hadoop Data Load Balancing Algorithm: described the Hadoop load balancing algorithm in detail. As this algorithm cannot balance the overload racks preferentially, which could lead to these racks breakdown, an improved algorithm propose to balance overload racks as soon as possible. According to different conditions, perform different operations:

- If of rack j, this rack is deleted from PriorBalanceList.
- If of rack k, this rack is deleted from ForBalanceList.
- If the PriorBalanceList is empty, the algorithm terminates and jumps to next step.
- If the ForBalanceList is empty, the algorithm terminates. Otherwise, the algorithm turns to next step → Moving data between PriorBalanceList list and Next ForBalanceList list [2]

Weighted round robin load balancing: (WRR) is a common routing policy offered in cloud load balancers. However, there is a lack of effective mechanisms to decide the weights assigned to each server to achieve overall optimal revenue of the system. The relations between probabilistic routing (PR) and weighted round robin (WRR) policies and introduce the result of the algorithms under different number of users classes. [3]

Autonomous Agent Based Load Balancing Algorithm (A2LB): A2LB which provides dynamic load balancing for cloud environment. The proposed mechanism has been implemented and found to provide satisfactory results. Algorithms of various agents deployed in proposed framework are given below respectively:

Migration Agent (MA): These agents are initiated by channel agent. It will move to other data centres and communicate with load agent of that data centre to enquire the status of VMs present there, looking for the desired configuration.

*Load Agent (LA):*It controls information policy and maintains all detail of a data centre. The major work of a load agent is to calculate the load on every available virtual machine after allocation of a new job in the data centre.

Channel Agent (CA): It controls the transfer policy, selection policy and location policy. On receiving the request from load agent, the channel agent will initiate some migration agents to other data centres for searching the virtual machines having similar configuration. [4]

Anonymous attribute-based privilege control scheme: This paper proposes a semi-anonymous attribute-based privilege control scheme *AnonyControl* and a fully-anonymous attribute-based privilege control scheme *AnonyControl-F* to address the user privacy problem in a cloud storage server. Using multiple authorities in the cloud computing system, proposed schemes achieve not only fine-grained privilege control but also identity anonymity while conducting privilege control based on users' identity information.[5]

LB-BC (Load Balancing based on Bayes and Clustering): LB-BC first has narrowed down the search scope by comparing performance values. Then, LB-BC has utilized Bayes theorem to obtain the posteriori probability values of all candidate physical hosts. Finally, LB-BC has combined probability theorem and the clustering idea to pick out the optimal hosts set, where these physical hosts have the most remaining computing power currently, for deploying and executing tasks by selecting the physical host with the maximum posteriori probability value as the clustering center and thus to achieve the load balancing effect from the long-term perspective. [6]

V. ANALYSIS AND DISCUSSION

Phantom replaces instances if they crash and terminates them based on rebalancing policies. Several rebalancing policies also propose that guide the deployments towards requested multi-cloud configurations while having minimal impact on the workload, if possible. [1] LB-BC proposes for the long-term load balancing effect and it has employed a heuristic idea based on Bayes theorem and the clustering process.[6] *Anonymous attribute-based privilege control scheme* to address the user privacy problem in cloud and achieve not only fine-grained privilege control.[5] *Comparison Between algorithms discussed is given below:*

Algorithm/ Method Used	***Advantages***	***Disadvantages***
Multicloud environment	aggressive policy provides fastest convergence	6.6% workload overhead due to premature

	time and the lowest excess cost, reducing it by factor of 3 over opportunistic policy only introducing	job termination required for immediate rebalancing.
Hadoop data load balancing algorithm	Total time used by Hadoop algorithm is shorter . This is due to different data computing and moving strategy.	But comparing the experimental results cannot conclude whose total time used for balance is much shorter.
Weighted round robin algorithm	1.OPT returns the best revenue *2.* Heuristic gives better result than simple weight setting. 3.Optimization program returns the best revenue among all the policies. And the revenue is around 27% higher than others.	1.There is a lack of effective mechanisms to decide the weight assigned to each server to achieve an overall optimal revenue of the system. 2.The heuristic algorithm requires only the demands but the problem is that it only works best for heavy load.
Anonymous attribute-based privilege control scheme	*AnonyControl-F* presented which fully prevents the identity leakage and achieve the full anonymity.	The authorities are well protected servers, it is hard to compromise even one authority, and the probability of compromising enough authorities to illegally decrypt some ciphertext is very low.
Autonomous Agent Based Load Balancing Algorithm (A2LB)	This mechanism is proactive load calculation of VM in a DC and whenever load of a VM reaches near threshold value, load agent initiates search for a candidate VM from other DCs.	A2LB takes optimum time when virtual machine becomes overloaded.
Heuristic approach for efficient task placement : LB-BC	1.Reduce the failure number of task deployment events obviously, improved throughput, and optimized the external services performance of cloud data centers.	1.With time increasing, its external service performance is not stable and has a waving mode. 2.Also, the external service performance of DLB is relatively better than that of LB-BC at initial time.

Table 1: Advantages and disadvantages

VI. PROPOSED METHODOLOGY

The proposed a load balancing algorithm which will transfer the load to another server in cloud when the current server is overloaded. When the multiple request are arrived to allocate the resource at server in cloud environment, the server gets overloaded at some instance. Kerberos authentication is used to validate the client-side credentials. This means that the client must request a Service Ticket valid for the Hadoop environment and submit this Service Ticket as part of the client connection. Validation is provided by a trusted third party in the form of the Kerberos Key Distribution Center.

HDFS Authentication:

Authentication process for HDFS is described below:

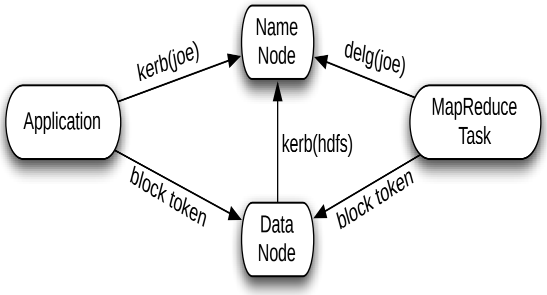

Fig. Authentication process of HDFS

- Clients authenticate to NameNode via:
  - Kerberos
  - Delegation tokens
- Client demonstrates authorization to DataNode via block access token
- DataNode authenticates to NameNode via Kerberos

MapReduce Authentication:

Authentication process for MapReduce is described below:

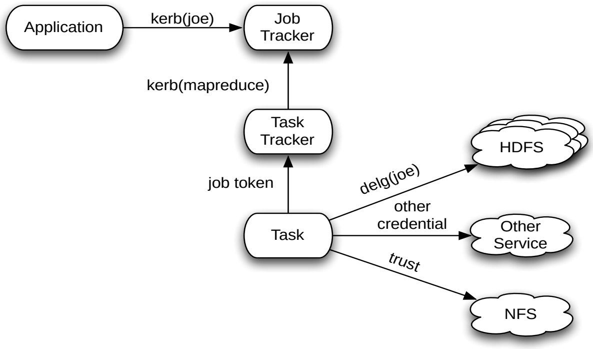

Fig.2 MapReduce Authentication

The process of authentication carried out as shown in fig. and steps invoved are-

- Client authenticates to JobTracker via Kerberos
- TaskTracker authenticates to JobTracker via Kerberos
- Task authenticates to the TaskTrackers using the job token
- Task authenticates to HDFS using a delegation token
- NFS is not Kerberized.

Hadoop and Kerberos Authentication Flow:

The process flow for Kerberos and Hadoop authentication is given in the below.

1. User obtains Ticket Granting Ticket (TGT)
2.Client application uses TGT to request a Service Ticket for Hadoop Service (HDFS/HIVE)
3. Client Application connects to Hadoop Service providing the Service Ticket to authentication
4. User authenticated using the Service Ticket & Service Key
5. Results from Hadoop Service

The first step, where the end user obtains a Ticket-Granting Ticket (TGT), does not necessarily occur immediately before the second step where the Service Tickets are requested. There are different mechanisms that can be used to obtain the TGT. Some users run a kinit command after accessing the machine running the Hadoop clients. Others integrate the Kerberos configuration in the host operating system setup. In this case, the action of logging on to the machine that runs the Hadoop clients will generate the TGT.

After the user has a Ticket-Granting Ticket, the client application access to Hadoop Services initiates a request for the Service Ticket (ST) that corresponds to the Hadoop Service the user is accessing. The ST is then sent as part of the connection to the Hadoop Service. The corresponding Hadoop Service must then authenticate the user by decrypting the ST using the Service Key exchanged with the Kerberos Key Distribution Center. If this decryption is successful the end user is authenticated to the Hadoop Service.

Their main merits are: 1) The proposed schemes are able to protect user's privacy against each single authority. 2) The proposed schemes are tolerant against authority compromise, analysis on security and performance provided to show feasibility of the scheme.

VII. EXPECTED RESULTS

In this paper, the developed cloud framework not only justified its optimum function in terms of minimization of execution or response time but also providing a secure data storage facility for public cloud infrastructure. In order to accomplish the goal of secure and authenticated data storage on public cloud, key distribution center technique has been implemented through enabling authentication using Kerberos, encryption using SSL/TLS, and authorization using HDFS.

VIII. CONCLUSION

It is important to evaluate solutions for cloud balancing implementations with an eye toward support for the needs of an actual IT department. Combining high availability with security is just as important. When the organization is using a network that's not its own for mission-critical application delivery, stability and security become paramount. Cloud computing has introduced a cost-effective alternative to building out secondary or even tertiary data centers as a means to improve application performance, assure application availability, and implement a strategic disaster recovery.

IX. FUTURE SCOPE

This paper has discussed secure load balancing that can be applied to clouds, but there are still other approaches that can be applied to balance the load in clouds. The performance of the given load balancer in Hadoop can also be increased by. In future, the plan is to conduct more experiments with various data sets in the future.

REFERENCES

[1] Dmitry Duplyakin, Paul Marshall Kate , Keahey
Henry Tufo,Ali Alzabarah,"Rebalancingin a Multi Cloud Environment", *Science Cloud '13,Proceedings of the 4th ACM WORKSHOP ON SCIENTIFIC CLOUD COMPUTING*, pp. 21-28, June 17, 2013.

[2] Kun Liu, Gaochao Xu and Jun'e Yuan, "An Improved Hadoop Data Load Balancing Algorithm", *JOURNAL OF NETWORKS*, VOL.8, NO.12, pp.2816-2822, December 2013.

[3] Weikun Wang, Giuliano Casale, "Evaluating Weighted Round Robin Load Balancing for Cloud Web Services",2014 16th *INTERNATIONAL SYMPOSIUM ON SYMBOLIC AND NUMERIC ALGORITHMS FOR SCIENTIFIC COMPUTING published in IEEE*, pp. 393 - 400 , 22-25 Sept. 2014 , 978-1-4799-8448-0/15 $31.00 © 2015 IEEE, doi :10.1109/SYNASC.2014.59

[4] Taeho Jung, Xiang-Yang Li, , Zhiguo Wan, and Meng Wan, "Control Cloud Data Access Privilege and Anonymity With Fully Anonymous Attribute-Based Encryption", *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY,* pp.190- 199 vol. 10, no. 1, January 2015

[5] Aarti Singh, Dimple Juneja, Manisha Malhotra , "Autonomous Agent Based Load Balancing Algorithm in Cloud Computing", *International Conference On ADVANCED COMPUTING TECHNOLOGIES AND APPLICATIONS (ICACTA-2015)*, 1877-0509 © 2015,

Published by Elsevier B.V. doi: 10.1016/ j .procs.2015.03.168

[6] Jia Zhao, Kun Yang, Xiaohui Wei, Yan Ding, Liang Hu, Gaochao Xu, "A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment", *IEEE TRANSACTIONS ON PARALLEL & DISTRIBUTED SYSTEMS*, vol.27, no. 2, pp. 305-316, Feb. 2016, doi:10.1109/TPDS.2015.2402655