

Volume 5, Issue 2, February 2015 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Review of Load Balancing for Distributed Systems in Cloud

Radha G. Dobale^{*}, Prof. R. P. Sonar Department of Computer Science and Engineering

Amravati University, Maharashtra, India

Abstract- Proposed work discusses the load balancing concept in a distributed manner in which nodes perform their load balancing tasks independently without synchronization or global knowledge regarding the system. A fully distributed load balancing algorithm is proposed to cope with the load imbalance problem. Instead of partitioning a file into a no. of chunks and balancing a load by migrating different chunks to different chunk servers, the load balance Nearest Search algorithm migrate one user's one whole file into any one node. Load is transferred from heavily loaded node to physically closed lightly loaded node. This proposal strives to balance the loads of nodes and reduce the demanded movement cost with reduce spending on technology as much as possible.

Keywords: distributed load balancing, Nearest Search algorithm, threshold load, Cloud storage capacity.

I. INTRODUCTION

With the exponential rise in the demand of clients worldwide, a large scale distributed systems have been introduced as a computing environment.

Cloud computing:

The definition of cloud computing provided by National Institute of Standards and Technology (NIST) says that: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, data storage, software applications and other computing services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cloud computing is an emerging computing technology that uses the Internet and central remote servers to maintain data and applications. Some applications are as Shown in fig 1.



Fig 1: Applications of cloud computing

There are three services of cloud shown in fig 2:

- 1) Infrastructure as a service (IaaS)
- 2) Software as a service (SaaS)
- 3) Platform as a service (PaaS)

To know about popular versions of cloud offerings are Software-as-a-Service (SaaS) and Infrastructure-as-a-Service (IaaS). With SaaS, the cloud service provider hosts your enterprise applications and associated data on its servers and storage systems. Users gain access to SaaS applications using a Web browser. And your company would typically pay a fee per user per month. With IaaS, the provider offers virtual machines, physical servers, storage, switching, and connectivity resources to run your enterprise applications on a pay-as-you-go basis. Platform as a service (PaaS) provides computing platform and a solution stack as a service. PaaS include facilities for application design, application development, testing and deployment as well as services such as team collaboration, web service integration, security, storage, state management etc.



Fig 2: Cloud services

Distributed file system:

Distributed file system for cloud is a file system that allows many clients to have access to the same data/file providing important operations (create, delete, modify, read and write). Each file may be partitioned into several parts called chunks. Each chunk is stored in remote machines. Typically, data is stored in files in a hierarchical tree where the nodes represent the directories. Hence, it facilitates the parallel execution of applications. There are several ways to share files in a distributed architecture.

Meanwhile, the security of the system must be ensured. Confidentiality, availability and integrity are the main keys for a secure system. Nowadays, users can share resources from any computer/device, anywhere and everywhere through internet thanks to cloud computing which is typically characterized by the scalable and elastic resources such as physical servers, applications and any services that are virtualized and allocated dynamically. Thus, synchronization is required to make sure that all devices are update. Distributed file systems enable also many big, medium and small enterprises to store and access their remote data exactly as they do locally, facilitating the use of variable resources.

Load balancing:

Load balancing is essential for efficient operations in distributed environments. It means distributing the amount of work to do between different servers in order to get more work done in the same amount of time and serve clients faster. In this case, consider a large-scale distributed file system. The system contains N chunkservers in a cloud (N can be 1000, 10000, or more), where a certain number of files are stored. Each file is split into several parts or chunks of fixed size (for example 64 megabytes). The load of each chunk server is proportional to the number of chunks hosted by the server.

II. MOTIVATION AND RELATED WORK

Cloud computing is a vast concept. Many algorithms have been proposed for load balancing in cloud computing. Some of those algorithms have been studied in this paper.

Paper [1] addresses proposed parallel hybrid dataflow architecture, a scalable dynamic load balancing circuit for the proposed architecture, and performance analysis. This deals with real-time applications of the architecture. The dynamic load balancing mechanism prevents excessive queuing of data and commands at a node during run-time and in doing such it balances the load over the entire system. The focus of this paper was three fold: first, on presenting the framework of the proposed HDCA system; second, on the modeling, design, and performance of a "basic" and then "modular" (scalable) dynamic load balancing circuit for a HDCA type computer system; and third, performance evaluation of a HDCA employing the dynamic load balancing circuit via a developed simulation.

Paper [2] presents a new threshold load balancing method for workstations which process the jobs with relatively long execution times. The new method decides a proper time to perform load balancing and does perform the balancing right after the detection of the load imbalance. In dynamic state, it performs static load balancing with a long fixed period regardless of what the value of the average idle-time may be so that we can avoid the load balancing overheads.

As stated in paper [3] Chord maintains its routing information as nodes join and leave the system. Attractive features of Chord include its simplicity, provable correctness, and provable performance even in the face of concurrent node arrivals and departures. Chord is a valuable component for peer-to-peer large-scale distributed applications such as cooperative file sharing, time-shared available storage systems, and distributed indices for document and service discovery, and large-scale distributed computing platforms.

Paper [4] focuses on fixing the number of balancing instants and optimizing the completion time over the strength of load balancing, which is controlled by the so-called gain parameter, with the double load-balancing strategy the overall completion time is further reduced in comparison to the single load balancing case. It is also seen that the optimal choice of the gain parameter depends on the delay and this dependence becomes more significant as the delays increase. In the ideal case where the communication and load-transfer delays are negligible and the time required to implement the load-balancing policy is also negligible, the best performance is obtained when the load balancing is executed almost continuously without any reservation. With a double-load balancing strategy, it is possible to achieve improved overall

performance, measured by the completion time of the total tasks in the system, in comparison to the single loadbalancing strategy. In either case, a performance almost comparable to the continuous-load-balancing strategy can be achieved.

In paper [5], it presents three contributions to research on middleware load balancing. Cygnus has an ability to make load balancing decisions based on application defined load metrics, dynamically (re)configure load balancing strategies at run-time, and transparently add load balancing support to client and server applications. A load balancing service adapts its load balancing decisions based on dynamic load changes. A load balancing service provide scalability to a distributed application by utilizing available computing resources to handle a large number of client requests and manage many servers efficiently. Load balancing middleware is an important technology for improving the scalability of distributed applications. This paper describes Cygnus, which is a middleware load balancing and monitoring service designed to satisfy the requirements. Cygnus provides a framework for integrating a range of adaptive and nonadaptive load balancing strategies, such as Round Robin, Random, Least Loaded, and Load Minimum, to help increase overall system scalability for many types of CORBA middleware-based distributed applications in an efficient, transparent, and portable manner.

In [6] paper, this presents a proximity-aware load balancing scheme by using the concept of virtual servers. The goals of this scheme are not only to ensure fair load distribution over nodes proportional to their capacities, but also to minimize the load-balancing cost (e.g., bandwidth consumption due to load movement) by transferring virtual servers (or loads) between heavily loaded nodes and lightly loaded nodes in a proximity-aware fashion. Load balancing scheme bearing network proximity in mind can reduce the bandwidth consumption (e.g., bisection backbone bandwidth) dedicated to load movement. Second, it can avoid transferring loads across high-latency wide area links, thereby enabling fast convergence on load balance and quick response to load imbalance. The proximity information used in work is to make load balancing fast and efficient. The goal of load balancing scheme is to align load distribution and node capacity inherent in P2P systems to ensure fair load distribution among nodes—that is, have nodes carry loads proportional to their capacities.

A histogram manager from paper [7] maintains a histogram that reflects a global view of the distribution of the load in the system, and a load-balancing manager that redistributes the load whenever the node becomes overloaded or underloaded. Advantage of this paper is, it reduces the cost of constructing histogram and Reduce the cost of maintaining histogram also reduce the cost of updating histogram. Each node in HiGLOB maintains the load information of nodes in the systems using histograms. HiGLOB enabled systems are superior over other methods.

Paper [8] introduce a load rebalancing algorithm to solve the load balancing problem among all chunk servers(compute nodes) in the distributed file system; it also ensures that one chunk(file block) of a file and its two copies are stored in three different chunk servers at the same time. File system can be stored in multiple machines which share and transfer data through the network with strong robustness and scalability and ensures the reliability of the system. Load rebalancing is one of the effective methods improving the overall performance of the system. A load rebalancing algorithm not only to achieve load balancing but also ensure the high reliability of the system.

The basic idea of hierarchical strategy in paper [9] is to divide processors into independent autonomous groups and to organize the groups in a hierarchy, thereby decentralizing the load balancing task. This deal with scalability challenges of load balancing at very large scale. reduces the time and memory required for load balancing hierarchical approach does not significantly compromise the quality of load balance achieved, hierarchical load balancing scheme is to focus on the optimizations that reduce communication, minimize memory usage, and limit data migration. Automatic dynamic hierarchical load balancing method that overcomes the scalability challenges of centralized schemes and poor solutions of traditional distributed schemes.

In paper [10], they improve on existing algorithms for item based active load balancing by relying on approximations of global properties. This algorithm reduces the network traffic induced by load balancing while achieving a better load balance than standard algorithms. By augmenting local knowledge with estimations of global aggregates, it is possible to reduce the total cost of load balancing. Load balancing algorithms have two goals: (a) improving the load distribution fairness and (b) minimizing the data moved around for achieving the first goal. This paper shows the benefits of adding global estimates for both, active and passive load balancing algorithms.

Paper [11] proposes a cloud task scheduling policy based on Load Balancing Ant Colony Optimization (LBACO) algorithm. The main contribution of our work is to balance the entire system load while trying to minimizing the make span of a given tasks set. A good task scheduler adapts scheduling strategy to the changing environment and the types of tasks. This paper proposed a Load Balancing Ant Colony Optimization (LBACO) algorithm to find the optimal resource allocation for each task in the dynamic cloud system.

In [12] author proposes some common load-balancing tactics will be introduced in this paper, which include: round-robin, weighted round-robin, least-connection, weighted least connection and shortest expected delay. Cloud computing enables shared servers to provide resources, software and data for collaborative services on demand with high interoperability and scalability and better uses of the resources. This present a feasible resource-aware load balancing mechanism by using existing proven technologies to meet higher SLA and the return of investment as well. The contributions of this paper proposed two tactics where resource-fit is shown the best policy by simulation.

In order to avoid the system burden caused by duplicate data, paper [13] proposes novel data center management architecture: Index Name Server (INS), which integrates deduplication and access point selection optimization techniques to enhance the performance of the cloud storage system. With the attempt to attain load balancing, this uses several specific INS parameters to dynamically monitor the parameters, like IP information and busy level index, to

avoid the network congestion or the long waiting time during data transmission. INS improves the efficiency of the cloud storage system. The proposed INS data center management mechanism omits the scanning procedure of traditional backup and decreases the backup cost and establish efficient backup of all schemes and methods.

In [14] algorithm the ant care for every node they visit and record their data for future decision making. The ants can modify their paths upon encountering any obstacles in their path. The movements of these ants independently update a solution set. The Traversal of ants in this system is generally of two types:

1) Forward movements

2) Backward movements

This efficiently distribution of the load among the nodes and such that the ants never encounter a dead end for movements to nodes for building an optimum solution set. The main benefit of this approach lies in its detections of overloaded and underloaded nodes and thereby performing operations based on the identified nodes. This simplistic approach elegantly performs task of identification of nodes by the ants and tracing its path consequently in search of different types of nodes. The concepts of Ant colony optimizations have modified the concepts where forward and trailing pheromones are used according to their convenience.

In paper [15], author proposes a new model for distributed load balancing allocation of virtual machine in cloud data center using the TOPSIS method-one of the most efficient Multi Criteria Decision Making (MCDM) techniques. This method can find the most suitable PM(Physical Machine) in the data center for the migrated VMs(Virtual Machines). System can achieve better load balancing in a large-scale cloud computing environment with less VM migration. Each node in the data center runs a module of the VM monitor which observes the local resource usages of the node. If the local observations reveal an anomaly that the resources are over-utilized or under-utilized, it takes decision namely: 1) Which VM to migrate from the problematic PM.

2) Which PM to migrate the chosen VM to.

Concept of paper [16] is to design a load rebalancing algorithm to reallocate file chunks such that the chunks can be distributed to the system as uniformly as possible while reducing the movement cost as much as possible. Here, the movement cost is defined as the number of chunks migrated to balance the loads of the chunkservers. Advantage of this paper is to balance the loads of nodes and reduce the demanded movement cost as much as possible, while taking advantage of physical network locality and node heterogeneity. This proposal is comparable with the existing centralized approach and considerably outperforms the prior distributed algorithm in terms of load imbalance factor, movement cost, and algorithmic overhead.

Paper [17] presents a hybrid control strategy for load balancing. On the one hand the storage node cluster redistributes the load in its local range. On the other hand the system applies the overlapping structure to distribute the load to the global storage nodes by batch iteration approach. This reduces the overall response time and better balance the global load. The approach of dynamic load balancing fully utilizes system resources, and thus improves the system performance. Migration method effectively reduces the system load. This article proposes a hybrid control strategy for load balancing. The local applies the centralized control strategy to quickly redistribute the load. The global architecture uses batch iteration to distribute the load to the global.

In paper [18], authors illustrate and define the load rebalancing problem in cloud DFSs. We advocate file systems in clouds shall incorporate decentralized load rebalancing algorithms to eliminate the performance bottleneck and the single point of failure. The resources in a load-balanced cloud can be well utilized and provisioned, maximizing the performance of MapReduce based applications. The centralized approach simplifies the design and implementation of a distributed file system. This proposal is comparable to the centralized algorithm in the Hadoop HDFS production system and dramatically outperforms the competing distributed algorithm in terms of load imbalance factor, movement cost, and algorithmic overhead.

Paper [19] presents the analysis of three contemporary algorithms namely Round Robin, Equally Spread Current Execution Load (ESCE), Throttled Load Balancing in cloud analyst tool to resolve the issue of cloud load balancing as a preparation phase for new load balancing technique. This helps to enhance the overall cloud performance. This paper proposed a new VM load balancing algorithm: Weighted Signature based Load Balancing (WSLB) algorithm proposed to minimize the users response time.

Paper [20] mainly concerns with the load balancing of cloud datacenters to improve efficiency of the host machine and minimize number of active host machine to support green computing concept. This paper introduces a threshold based Dynamic compares and balance algorithm (DCABA) for cloud server optimization. DCABA minimizes the number of host machines to be powered on, for reducing the cost of cloud services. Our approach can serve the purpose of service cost reduction in cloud industry with effective utilization of available resources and reduce the operational cost of cloud application. This paper has shown the applicability of load balancing and server consolidation techniques to obtain measurable improvements in server workload management and minimize the cost of cloud services.

Paper [21] provides a comparative study between the three load balancing architectures in cloud computing: centralized, decentralized and hierarchical load balancers. Load balancers promote achieving three main objectives. First, improving overall system performance by reaching high resource utilization ratio. Second, avoiding system bottleneck that occurs due to load imbalance. Finally, achieving high providers and user's satisfaction by striving to increase the system throughput and decrease the job processing time. Cloud computing is a relatively new IT paradigm that offers huge amount of resources at reasonable cost. Load balancing solutions that are capable of maintaining low values for response time and server loads.

Paper [22] presents a load balanced co-location algorithm is incorporated into CoHadoop and it is termed as CoHadoop++. It achieves load balance in the cluster through optimal selection of data nodes based their load. CoHadoop++ tries to select nodes with lesser load and thus decreases the load imbalance caused due to co-location. File Partition Generator helps in maintaining the consistency of the input file so that it can be retrieved without any loss of data. CoHadoop++ ensures that the fault tolerance property of Hadoop is not compromised, when excluding nodes from the node selection policy.

Load rebalancing algorithm is implemented in paper [23], so that central node should not overload. The implementation is done in Hadoop distributed file system. As apache Hadoop is used, security issues are arises. To solve these security issues and to increase security, Kerberos authentication protocol is implemented to handle multiple nodes. This paper shows real time implementation experiment on cluster. Cloud computing has improved performance, reduced software cost, instant software updates, improved document format compatibility, unlimited storage capacity etc. As Hadoop's use and demand grew in the network, handle big data security became critical, So that authentication mechanism Kerberos is used. This approach has the advantage that one could continue to use the tokens to supplement a different primary authentication mechanism.

Paper [24] proposes a novel decentralized load balancing architecture, called tldlb (two-level decentralized load balancer). This distributed load balancer takes advantage of the decentralized architecture for providing scalability and high availability capabilities to service more cloud users. This is simple and efficient; it is well-suited for cloud computing environments to service more requests with less response time. This proposed a novel neural network-based load balancing algorithm, nn-dwrr, to distribute incoming requests to appropriate VMs.

The Central Load Balancer (CLB) in paper [25] is connected to all users and virtual machines present in cloud data center through Data center Controller. The Central Load Balancer calculates the priorities of virtual machines based on their CPU speed (MIPS) and memory. Load balancing helps to achieve a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource, minimizing resource consumption, implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning etc. In proposed Central Load Balancer (CLB) technique, authors tried to avoid the situation of over loading and under loading of virtual machines. The Central Load Balancer (CLB) manages load distribution among various virtual machines and assigns load corresponding to their priority and states. In this way this technique efficiently shares the load of user requests among various virtual machines.

In paper [26], authors propose two novel brownout-aware load balancing algorithms. To test their practical applicability, they extended the popular lightpd web server and load-balancer, thus obtaining a production-ready implementation. This algorithm improves user experience with high statistical significance, while preserving response time predictability. The adoption of the brownout paradigm allows the service to autonomously reduce computing capacity requirements by degrading user experience in order to guarantee that response times are bounded. This paper presents a novel approach for improving resilience, the ability to hide failures, in cloud services using a combination of brownout and load-balancing algorithms.

Paper [27] combined with greedy algorithm; the scheme provides a better load balancing algorithm for different load cases (CLB). CLB algorithm utilizes entropy and the scope of invalid cache invalid as the evaluation basis of load balancing effect. In this paper, a cache-invalidation-scope model based on the improved consistent hash algorithm is put forward. The most contribution of this paper is to improve the existing consistent hash algorithm and make it suitable for load balancing, besides, a cache invalidation-scope model is proposed providing a favorable load balancing scheme. The proposed algorithm lifts up the performance of key value caching system. In this paper, a new scheme of load balancing for key value cache system in cloud environment is proposed in consideration the effect of load balancing and the scope of invalid cache. Cache-invalidation-scope model is established to improve the effect of load balancing.

III. SUBJECT OF WORK

Files and nodes can be dynamically created, deleted and appended. This results in load imbalance in distributed file system. A fully distributed load balancing algorithm is proposed to cope with the load imbalance problem. Instead of partitioning a file into a no. of chunks and balancing a load by migrating different chunks to different chunk servers, the load balance Nearest Search algorithm migrate one user's one whole file into any one node. So that the time complexity of manipulating of hash addresses to keep track of these file chunks are avoided. By doing this, this paper eliminates previous time consuming procedure.

Higher capacity nodes carry more loads. Load is transferred from heavily loaded node to physically closed lightly loaded node. This method balances the load when it reaches to threshold/control line only. Where it treats the overloaded portion above the threshold line and underloaded portion as below the load control line i.e. threshold line.

IV. PROPOSED ALGORITHM

Step 1: Register user-

If Cloud storage capacity >= All nodes used storage capacity + User trying selected storage plan capacity. Notify user "Successfully registered".

Else notify user "Currently registration closed. Try after some time."

Step 2: Return to home page.

Step 3: Login user-

If user storage plan validity is not yet expired.

February - 2015, pp. 393-400

Else notify user "Your storage plan has expired".
Step 4: After Successful Login.
If validity date is after 7 days or less;
Notify user "Resubscribed a plan otherwise account will be deactivated".
Else check some conditions before file upload.
Step 5: If Total storage capacity of selected plan >= Used storage capacity + User trying file uploading capacity
Goto step 6 for Nearest Node First with Nearest Search Algorithm.
Else goto step 5.
Step 6: Notify user as "Insufficient storage space. You can expand capacity".
If expanded a storage capacity, goto step 4.
Else logout.
Step 7: For i=1 till i \leq =5 do i++
If nearest i^{th} node load is below the threshold line and <i>Node remaining storage capacity</i> >= User trying to
upload file capacity
Then upload file. Go to step 11.
Step 8: For $i=1$ till $i \le 5$ do $i++$
If nearest i^{th} Node remaining storage capacity >= User trying to upload file capacity
Then upload file. Go to step 11.
Step 9: X=Total no of nodes in cloud
For $i=6 \text{ till } i \leq X \text{ do } i++$
If Empty nodes available and Node storage capacity >= User trying to upload file capacity
Then upload file. Go to step 11.
Step 10: For i=6 till i $\leq X$ do i++
If nearest i^{th} Node storage capacity >= User trying to upload file capacity
Then upload file.
Step 11: Users login home page.

V. ADVANTAGES

- Maintenance: Maintenance of cloud computing applications is easier because they do not need to installed on each user's computer and can be accessed from different places.
- Load balancing: Load balancing distributing the amount of work to do between different servers in order to get more work done in the same amount of time and serve clients faster.
- Scalability: Load Control (balancing) allows cloud computing to scale up to increasing demands by efficiently allocating dynamic local work load across all nodes. No more infrastructure, investment or time spent adding new servers.
- Availability: Cloud load balancing reduces costs associated with document management systems and maximizes availability of resources.
- Mobility: Cloud mobility enables access anywhere with a web connection i.e. globalised the work.
- Reduce movement cost: By using Nearest Search Algorithm, the movement cost will be reduced.
- Flexibility: Users can access stored data anywhere in the world but the required thing is computer/laptop/smart phones (android) and other applicable devices with internet connection.
- Device and location independence: Enables users to access systems using a web browser regarding of their location or what device they use (e.g. PC, mobile phone).
- Performance: By reducing the movement cost, can achieve greater performance.
- Cost Efficiency: Cost can minimize with cloud services updating and managing software or applications i.e. cost can be reduce by spending on technology.
- Reduce capital cost: Reduces cost by not needing to buy updates and or newer versions of software and hardware.
- Monitor project more effectively: Can access project from any location with internet connection.
- Highly automated: No need to purchase updated software's, everything has been set up and ready to use. Here you can also cut an employee, who takes care of software updates.
- More storage capacity: No need to worry about your lot of data and files to store, this provides more data to save the files in server. Here depending upon the data and usage you can choose the plans, available in different modes. Everything is online, store your entire data in cloud and can access at any time in browser.
- Minimize response time: By using Nearest Search Algorithm, it minimizes the response time to store or upload/download/view the data.
- Maximize throughput: Because of reducing response time and maximizing the performance, it's possible to achieve greater throughput.

VI. CONCLUSION

Proposed work strives to balance the loads of nodes and reduce the demanded movement cost with reduce spending on technology as much as possible. Also optimize resources use, maximize throughput and minimize response time.

VII. FUTURE SCOPE

Energy efficiency has become one of the most active topics in large scale of data center or cloud computing environment today. In future work, intend to improve power consumption of such nodes which are underutilized by 25% by making group of such servers and computing load in a single server which should be less than 75% or specified threshold control line for Green Computing.

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and guidance of Prof. Rashmi P. Sonar. I also thanks to IEEE explorer of funding and carrying out literature review for the research.

REFERENCES

- [1] J. Robert Heath, Saivenkatesh Ramamoorthy, Charles E. Stroud and Andrew D. Hurt, *Modeling, Design, and Performance Analysis of a Parallel Hybrid DataCommand Driven Architecture System and Its Scalable Dynamic Load Balancing Circuit*, Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions, Jan 1997, Volume:44, Issue: 1, Pages: 22-40.
- [2] Hye-Seon Maeng, Hyoun-Su Lee, Tack-Don Han, Sung-Bong Yang, Shin-Dug Kim, *Dynamic Load Balancing of Iterative Data Parallel Problems on a workstation clustering*, High Performance Computing on the Information Superhighway, 1997. HPC Asia '97, Aril 1997, Pages: 563 567.
- [3] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan, *Chord A Scalable Peer to Peer Lookup Protocol For Internet Applications*, Networking, IEEE/ACM Transaction, Feb 2003, Volume:11, Issue:1, Pages:17-30.
- [4] S. Dhakal, B. S. Paskaleva, M. M. Hayat, E. Schamiloglu, C. T. Abdallah, *Dynamical Discrete-Time Load Balancing in Distributed Systems in the presence of time delays*, Decision and Control, 2003. Proceedings. 42nd IEEE Conference, Dec 2003, Volume:5, Pages:5128-5134.
- [5] Jaiganesh Balasubramanian, Douglas C. Schmidt, Lawrence Dowdy, and Ossama Othman, *Evaluating the Performance of Middleware Load Balancing Strategies*, Enterprise Distributed Object Computing Conference, 2004. EDOC 2004. Proceedings Eighth IEEE International, Sept 2004, Pages:135-146.
- [6] Yingw u Zhu and Yiming Hu, *Efficient, Proximity-Aware Load Balancing for DHT Based P2P Systems*, Parallel and Distributed Systems, IEEE Transaction, Aril 2005, Volume:16, Issue:4, Pages: 349-361.
- [7] Quang Hieu Vu, Member, IEEE, Beng Chin Ooi, Martin Rinard, and Kian-Lee Tan, *Histogram-Based Global Load Balancing in Structured Peer to Peer Systems*, Knowledge and Data Engineering, IEEE Transaction, April 2009, Volume:21, Issue:4, Pages:595-608.
- [8] Wenqiu Zeng, Ying Li, Jian Wu, Qingqing Zhong, Qi Zhang, *Load rebalancing in Large-Scale Distributed File System*, Information Science and Engineering (ICISE), 2009 1st International Conference, Dec 2009, Pages:265-269.
- [9] Gengbin Zheng, Esteban Meneses, Abhinav Bhatel'e and Laxmikant V. Kale, *Hierarchical Load Balancing for Charm++ Applications on Large Supercomputers*, Paralle processing workshops (ICPPW), 2010 39th International conference, Sept 2010, Pages: 436-444.
- [10] Nico Kruber, Mikael H^{*}ogqvist, Zuse-Institut Berlin, Takustr. Thorsten Schutt, *The Benefits of Estimated Global Information in DHT Load Balancing*, Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium, May 2011, Pages: 382-391.
- [11] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang, *Cloud Task scheduling based on Load Balancing Ant Colony Optimization*, Chinagrid Conference (ChinaGrid), 2011 Sixth Annual, August 2011, Pages:3-9.
- [12] Rich Lee, Bingchiang Jeng, *Load-Balancing Tactics in Cloud*, Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011 International Conference, Oct 2011, Pages:447-454
- [13] Tin-Yu Wu, Wei-Tsong Lee, Yu-San Lin, Yih-Sin Lin, Hung- Lin Chan, Jhih-Siang Huang, *Dynamic Load Balancing Mechanism based on Cload Storage*, Computing, Communications and Applications Conference, Jan 2012, Pages:102-106.
- [14] Kumar Nishant, Pratik Sharma, Vishal Krishna, Chhavi Gupta and Kuwar Pratap Singh, Nitin and Ravi Rastogi, *Load Balancing of Nodes in Cloud Using Ant Colony Optimization*, Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference, March 2012, Pages:3-8.
- [15] Fei Ma, Feng Liu and Zhen Liu, *Distributed Load Balancing Allocation of Virtual Machine in Cloud Data Center*, Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference, June 2012, Pages: 20-23.
- [16] Hung-Chang Hsiao, Hsueh-Yi Chung, Haiying Shen, Yu-Chang Chao, *Load Rebalancing for Distributed File Systems in Clouds*, Parallel and Distributed Systems, IEEE Transactions, June 2012, Volume:24, Issue: 5, Pages:951-962.
- [17] Yilim Lu, Jian Zhang, Shaochun Wu and Shujuan Zhang, *A Hybrid Dynamic Load Balancing Approach for Cloud Storage*, Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference, August 2012, Pages:1332-1335.

- [18] Hsueh-Yi Chung, Che-Wei Chang, Hung-Chang Hsiao, Yu-Chang Chao, *The Load Rebalancing Problem in Distributed File systems*, Cluster Computing (CLUSTER), 2012 IEEE International Conference, Sept 2012, Pages:117-125.
- [19] Mr. M. Ajit, Ms. G. Vidya, *VM Level Load Balancing in Cloud Environment*, Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference, July 2013, Pages:1-5.
- [20] Yatendra Sahu, Yatendra Sahu, Rajeev Kumar Gupta, *Cloud Server Optimization with Load Balancing and Green Computing Techniques Using Dynamic Compare and Balance Algorithm*, Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference, Sept 2013, Pages:527-531.
- [21] Ektemal Al-Rayis, Heba Kurdi, *Performance Analysis of Load Balancing Architectures in Cloud Computing*, Modelling Symposium (EMS), 2013 European, Nov 2013, Pages:520-524.
- [22] Nishanth S, Radhikaa B, Ragavendar T J, Chitra Babu, and Prabavathy B, *CoHadoop++ A Load Balanced Data Colocation in Hadoop Distributed File System*, Advanced Computing (ICoAC), 2013 Fifth International Conference, Dec 2013, Pages:100-105.
- [23] Ms. Vidya N. Chiwande, Prof. Animesh R. Tayal, *An Approach to Balance the Load with Security for Distributed File System in Cloud*, Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference, Jan 2014, Pages:267-270.
- [24] Chung-Cheng Li and Kuochen Wang, *An SLA aware load balancing scheme for cloud datacenters*, Information Networking (ICOIN), 2014 International Conference, Feb 2014, Pages:58-63.
- [25] Gulshan Soni, Mala Kalra, *A Novel Approach for Load Balancing in Cloud Data Center*, Advance Computing Conference (IACC), 2014 IEEE International , Feb 2014, Pages:807-812.
- [26] Cristian Klein, Alessandro Vittorio Papadopoulos, Manfred Dellkrantz, Jonas Durango, Martina Maggio, Karl-Erik Arzen, Francisco Hernandez-Rodriguez, Erik Elmroth, *Improving Cloud Service Resilience using Brownout-Aware Load-Balancing*, Reliable Distributed Systems (SRDS), 2014 IEEE 33rd International Symposium, Oct 2014, Pages:31-40.
- [27] Tao Wang, Xin Lv, Fang Yang, Wenhuan Zhou, Rongzhi Qi, HuaiZhi Su, A load balancing scheme for distributed key-value caching system in cloud environment, Distributed Computing and Applications to Business, Engineering and Science (DCABES), 2014 13th International Symposium, Nov 2014, Pages:63-67.