ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 5, Issue 3, March 2018

Real Time Load Balancing for Private and Public Virtual Cloud

[1] Srikant Bagewadi, [2] Vinod Alone, [3] Atul Shintre, [4] Manish Gangawane
[1] [2] [3] [4] Assistant Professor, Department of Computer Engineering, Padmabhushan Vasantdada Patil Pratishthan's College of Engineering, Mumbai, India

Abstract: - **Cloud computing is a internet based computing that delivers hardware, software and platform as a services to the user based on their demand at a very low cost, so today everyone is moving to the cloud, so the load on the cloud is also increasing, it has become essential to balance the load on cloud dynamically, however the existing system does not uses the system information to balance the load dynamically, so in these project we proposed a unique system that uses system information to balance the load dynamically for which we have introduced a better load balancing model for the private cloud based on the load partitioning using the clustering concept with the game theory mechanism to choose different strategy for the different clusters in a different situation which thereby improves the system performance and maintain the stability and also with the help of real time services which is deployed on a real private and public virtual cloud, where we show that the proposed system performs better in-terms of optimum utilization of resources, load equality and execution time.**

Keywords**: - Cloud Computing, Load Classification, k-means clustering, and Load Balancing.**

I. INTRODUCTION

Cloud computing is a internet based computing that delivers hardware, software and platform as a services to the user based on their demand at a very low cost. Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs in the cloud computing environment is a very complex problem with load balancing receiving much attention for researchers. Since the job arrival pattern is not predictable and the capacities of each node in the cloud differ, for load balancing problem, workload control is crucial to improve system performance and maintain stability. Load balancing schemes depending on whether the system dynamics are important can be both static and dynamic. Static schemes do not use the system information and are less complex while dynamic schemes will bring additional costs for the system but can change as the system status changes. A dynamic scheme is used here for its flexibility. The model has a main controller and balancers to gather and analyze the information. Thus, the dynamic control has little influence on the other working nodes. The system status then provides a basis for choosing the right load balancing strategy.

The load balancing model given in this project is aimed at the private and public virtual cloud, so this model divides the total load into smaller sub loads using the clustering concept such that the large size jobs are classified as one and whereas smaller size jobs are classified as another cluster. When the environment is very large and complex, these divisions simplify the load balancing. The cloud has a main controller that chooses the suitable cluster partitions for arriving jobs based on job size and also it polls the performance of Server/Node and determines the performance cost for that respective Server/Node, while the balancer for each cluster partition chooses the best load balancing strategy. The problem of the work can be defined as to provide Game Theory based Load Balancing technique coupled with scheduling that can provide better performance for both private and public virtual cloud and analyze the performance of the same to affirm the suitability of the method.

II. ORGANIZATION

Section I discusses the introduction, section III discusses the related work, section IV discusses the proposed system, section V discusses the methodology, section VI discusses the simulation and results, section VII discusses the conclusion, and the references.

III. RELATED WORK

There have been many studies of load balancing for the cloud environment. Load balancing in cloud computing was described in a white paper written by Adler [1] load balancing in the cloud. However, load balancing in the cloud is still a new problem that needs new architectures to adapt too many changes. Mladen A. Vouk [2] has discussed the concept of "cloud" computing, some of the issues it tries to address, related research topics, and a "cloud" implementation based on VCL technology. Neha Gohar Khan and Prof. V. B. Bhagat [3] has described the concepts of cloud computing and some of the methods of load balancing in large scale Cloud systems. Their aim is to provide an evaluation and comparative study of these approaches, demonstrating different algorithms for load balancing and to improve the different performance parameters like throughput, response time, latency etc. for the clouds. Bin Fan, Hyeontaek Lim, David G. Andersen, Michael Kaminsky [4] have demonstrated that a small, fast front-end cache that can ensure effective load-balancing, regardless of the query distribution and they have proved a lower bound on the cache size that depends only on the number of back-end nodes in the system, not the number of items stored. Wilhelm Kleiminger, Evangelia Kalyvianaki, Peter Pietzuch [5] have presented a combined stream processing system that, as the input stream rate varies, adaptively balances workload between a dedicated local stream processor and a cloud stream processor. This approach only utilizes cloud machines when the local stream processor becomes overloaded. Nidhi Jain Kansal and Inderveer Chana [6] have presented a systematic review of existing load balancing techniques. Out of 3,494 papers analyzed, 15 papers are identified reporting on 17 load balancing techniques in cloud computing. This study concludes that all the existing techniques mainly focus on reducing associated overhead, service response time and improving performance etc. Various parameters are also identified, and these are used to compare the existing techniques. S. Mohana Priya, B. Subramani [7] has proposed new load balancing algorithm for the virtual machines and a task scheduling algorithm. Their experimental result show that if an efficient virtual machine is selected for process and minimum execution time of task, it increases the performance and decreases the average response time and cost in cloud network. Branko Radojevic, Mario Zagar [8] have presented a new algorithm that incorporates information from virtualized computer environments and end user experience in order to be able to proactively influence load balancing decisions or reactively change decision in handling critical situations. Soumya Ray and Ajanta De Sarkar [9] have presented a review of a few load balancing algorithms or technique in cloud computing. Their objective is to identify qualitative components for simulation in cloud environment and then based on these components, execution analysis of load balancing algorithms are also presented. Abhijit A. Rajguru, S.S. Apte [10] have presented the performance analysis of various load balancing algorithms based on different parameters, considering two load balancing approaches static and dynamic. Illa Pavan Kumar,Subrahmanyam Kodukula [11] have discussed the issues and limitations of conventional load balancing techniques if deployed in cloud, importance of building scalable architecture for cloud by exploring scalability solutions provided by cloud vendors.

IV. PROPOSED SYSTEM

A. Proposed System

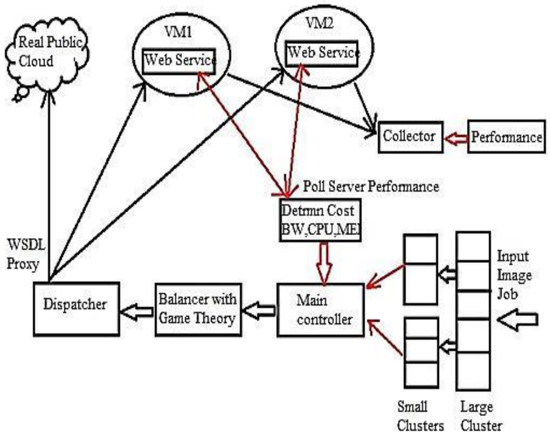


Figure 1: Architecture of the proposed system

B. Algorithm

Step 1: Input the image folder that containing the image jobs of different size.

Step 2: Classify the images into different group using the K-Mean s Clustering Algorithm.

Step 3: Compute moment job load and mean job load.

Step 4: Polls the Server/Node Performance and Determine the Cost for the respective Server/Node.

Step 5: Check which of the Server/Node is Overloaded and Under Loaded based on step 4.

Step 6: Check which of the Cluster is having small and large size image job based on step 3.

Step 7: Schedule small size image job to Heavily Loaded Server/Node i.e. remote host.

Step 8: Schedule large size image job to Lightly Loaded Server/Node i.e. local host.

Step 9: Repeat step 3 to step 8 until all jobs get done.

V. METHODOLOGY

A. Load Partitioning

There are different types of cloud are available where we have mainly focused our work towards private and public cloud. In Private Cloud, the cloud infrastructure is owned and managed by single organization and services are accessible only for that specific organization. In public cloud, the cloud infrastructure is owned and managed by cloud service provider and services are accessible for the public across the world wide at very low cost.

So therefore today everyone is moving to cloud because of which the load on cloud is increasing very rapidly and it has became essential to balance the load dynamically over the cloud. So it has become very important to classify the load properly based on the load size such that it minimizes the execution time and improves the system performance etc.

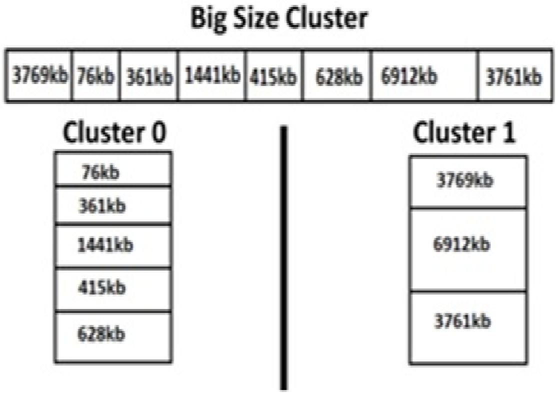


Figure 2: Example of Load Partitioning

As shown in fig 2, Load partitioning is used to manage a large load on the private and public cloud. A Load partitioning is a process of dividing or classifying big size load into smaller size load based on the Clustering technique. Where large size cluster is partitioned into smaller sub-size clusters such that large size jobs are classified as one cluster and the smaller size jobs are classified as another cluster by using the K-Means Clustering algorithm [15].

VI. RELATIONSHIP BETWEEN MAIN CONTROLLER, BALANCER, DISPATCHER AND NODES.

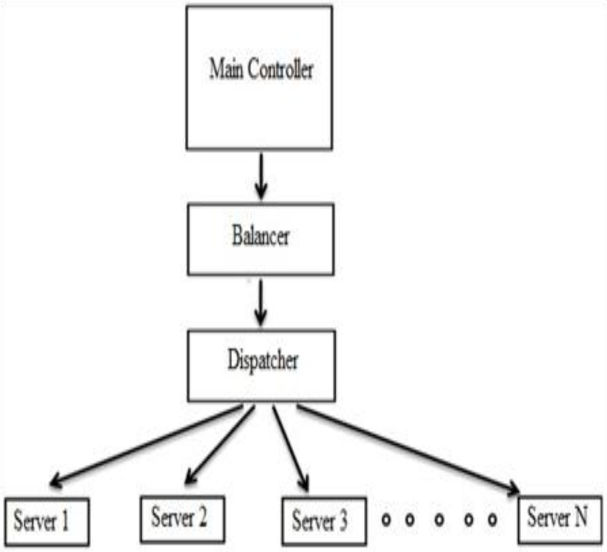


Figure 3: Relationship between Main Controller, Balancer, Dispatcher and Nodes/Servers.

A. Main Controller

Main controller creates the cluster partitions based on the clustering concept using the k-means clustering algorithm. Where it classifies the jobs into the different cluster based on the file/job size between the different job such that the large size jobs are classified as one cluster and smaller size jobs are classified as another cluster and then it invokes the real time services, which are deployed on local host and real private and public virtual cloud i.e. remote hosts, using which it polls the performances of Server/Node such as availability of Server/Node memory, bandwidth, latency, CPU utilization etc. based on which it estimates the Server/Node cost which is nothing but it determines the amount of load over that specific Server/Node using the below given formula.

Server_Cost = (CPU_Usage/100) + (Mem/100) + (1/(BW/100)) + (Latency/100)

B. Balancer

As now the balancer is having with the information of no of clusters and cost of the different Server/Node, which is determined by the main controller, by using these information balancer makes the decision based on the game theory concept that which of the Server/Node is heavily loaded and which of the Server/Node is lightly loaded and also makes the decision about which of the cluster job load is of smaller and larger size, depending upon which it assigns the scores to the Server/Node and depending upon these scores of the Server/Node, it schedules the large size jobs of the cluster to the Server/Node which is lightly loaded and small size jobs of the cluster to the Server/Node which is heavily loaded, such that it avoids the overloading of single Server/Node and also it performs better in-terms of optimum utilization of resources, load equality and execution time.

C. Dispatcher

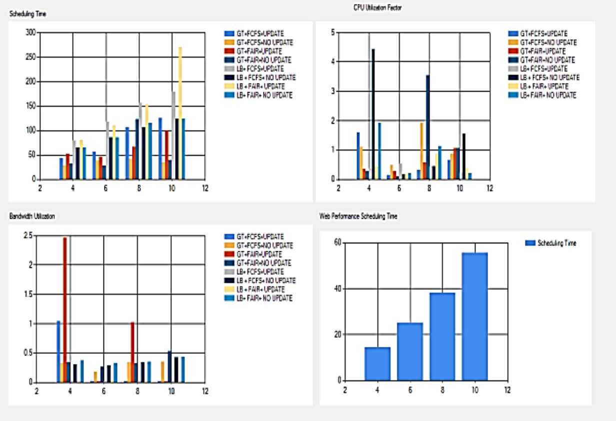
It just fetches and starts executing the jobs from the appropriate cluster to the appropriate Server/Node which are already been scheduled by the balancer for execution and also parallel keeps on updating the server performance to maintain stability over the cloud.

VII. SIMULATION AND RESULTS

We have simulated this work on two Server/Node such that one Server/Node is configured as local host and other Server/Node is configured as remote host and we have deployed our web service over these two Server/Node where it keeps on polling the performance of the respective Server/Node such as Bandwidth Utilized, CPU Utilized, Occupied Memory on Server/Node and Latency taken etc. based on which Cost of the respective Server/Node is calculated using which we determine which of the Server/Node is under loaded and overloaded and based on which the jobs are scheduled and executed over that respective Server/Node to optimize the resource utilization, minimize the execution time and improve the system performance. Here "UPDATE" refers to as polling Server/Node performance periodically in between dispatching process.

Table1: Abbreviations

GT+FCFS+UPDATE	Game Theory With FCFS via Update
GT+FCFS+NO UPDATE	Game Theory With FCFS via No Update
GT+FAIR+UPDATE	Game Theory With FAIR via Update
GT+FAIR+NO UPDATE	Game Theory With FAIR via No Update
LB+FCFS+UPDATE	Non Game Theory With FCFS via Update
LB+FCFS+NO UPDATE	Non Game Theory With FCFS via No Update
LB+FAIR+UPDATE	Non Game Theory With FAIR via Update
LB+FAIR+NO UPDATE	Non Game Theory With FAIR via No Update



A. Performance Graph

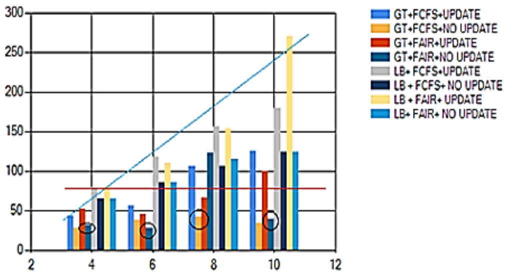


Figure 4: Performance Graph for No of Jobs v/s Scheduling Time

Fig 4, Shows that Game Theory based strategy performs better than usual cost based scheduling. In three iterations Fair Scheduling with Game theory where performance is not polled frequently has performed better than the rest. This proves that once a perfect limit is adopted for performance update, the system can produce optimal result. The work also shows that the response time of the proposed work does not increase linearly with increasing jobs. Hence it can be suitably adopted for real time time-critical applications.

B. Bandwidth Utilization graph

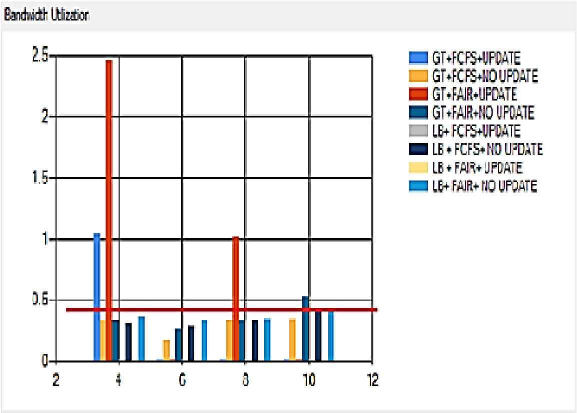


Figure 5: Bandwidth Utilization Graph

Without a proper load balancing mechanism, required bandwidth would also have been increasing with increasing jobs. Hence it can be safely said that updated performance matrix is not a prerequisite for dispatching every job. Also it can be seen that the proposed technique has lowest bandwidth utilization factor.

C. CPU Utilization Graph

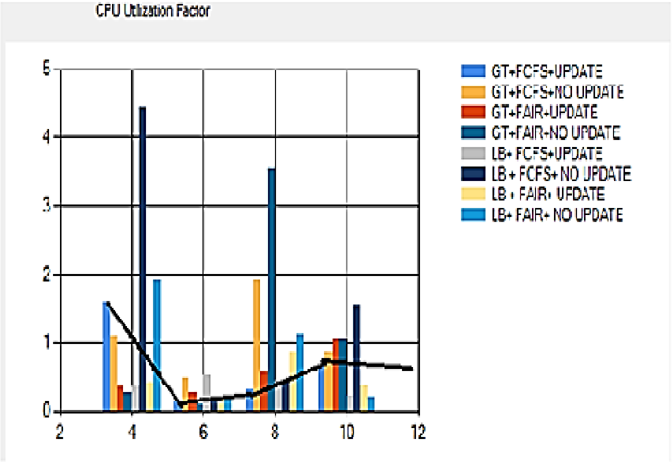


Figure 6: CPU Utilization Graph

CPU utilization proves that by using appropriate load balancing technique, jobs can be prevented from fetching excessive memory and resources.

D. Web Performance Graph

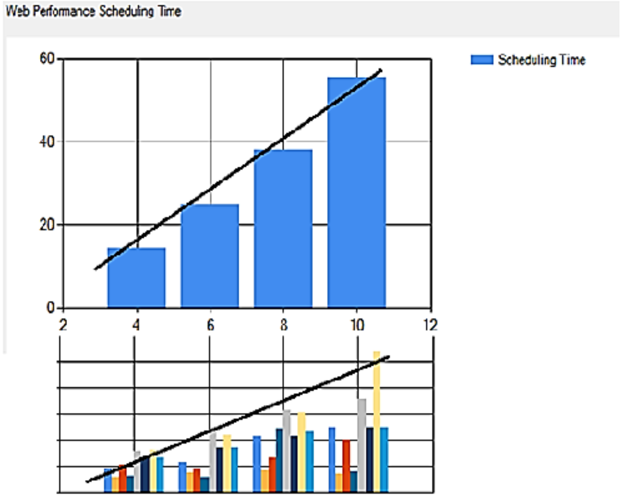


Figure 7: Web Performance Graph for No of Jobs v/s Scheduling Time

Web Performance analysis when compared with the simulation result the same trend. Therefore it can be said that the simulation results can be closely considered in the same matrix as that of real time result. Hence the proposed technique can be adopted in real time load scheduling and balancing for cloud.

VIII. CONCLUSION

Load balancing in a cloud is most important aspect with the corresponding to smooth and optimum user services, most sophisticated load balancing technique adopts different algorithms ranging from scheduling based balancing, cost based balancing, hybrid technique, however these technique do not combine the user level load with server optimization, so in these work we adopted a unique mechanism of game theory which is combined with the FCFS and FAIR scheduling technique to provide optimized services to the clients as well as to the server, we have demonstrated the concept by implementing the services in a real private and public virtual cloud and the result proves that the proposed system can efficiently optimize the user performance and as well as balance server side load.

REFERENCE

[1] B. Adler, Load balancing in the cloud: Tools, tips and techniques,http://www.rightscale.com/infocenter/whitepapers/Load-Balancing-in-the-Cloud.pdf, 2012.

[2] Mladen A. Vouk, Cloud Computing – Issues, Research and Implementations, http://hrcak.srce.hr/file/69202.

[3] Neha Gohar Khan Prof. V. B. Bhagat, An Systematic Overview On Cloud Computing and Load Balancing in the Cloud, ISSN: 2278-0181 Vol. 2 Issue 11, November - 2013

[4] Bin Fan, Hyeontaek Lim, David G. Andersen, Michael Kaminsky, Small Cache, and Big Effect: Provable Load Balancing for Randomly Partitioned Cluster Services, http://www.cs.cmu.edu/~hl/papers/loadbal-socc2011.pdf.

[5] Wilhelm Kleiminger, Evangelia Kalyvianaki, Peter Pietzuch, Balancing Load in Stream Processing with the Cloud,http://www.vs.inf.ethz.ch/publ/papers/Wilhelm%20Kleiminger-balanc-2011.pdf

[6] Nidhi Jain Kansal and Inderveer Chana, Existing Load Balancing Techniques In Cloud Computing: A Systematic Review, http://www.bioinfo.in/contents.php?id=45.

[7] S. Mohana Priya, B. Subramani, A New Approach For Load Balancing In Cloud Computing, ISSN:2319-7242 Volume 2 Issue 5 May, 2013

[8] Branko Radojevic, Mario Zagar, Analysis of Issues with Load Balancing Algorithms in Hosted (Cloud) Environments, MIPRO, 2011 Proceedings of the 34th International Convention, ISBN: 978-1-4577-0996-8.

[9] Soumya Ray and Ajanta De Sarkar, Execution Analysis of Load Balancing Algorithms in Cloud Computing Environment, http://airccse.org/journal/ ijccsa/papers/2512ijccsa01.pdf.

[10] Abhijit A. Rajguru, S.S. Apte, A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters, (IJRTE)ISSN: 2277-3878, Volume-1, Issue-3, August 2012.

[11] Illa Pavan Kumar,Subrahmanyam Kodukula, A Generalized Framework for Building Scalable Load Balancing Architectures in the Cloud, (IJCSIT) Vol. 3 (1) , 2012, 3015 – 3021

[12] Gerald Sabin Garima Kochhar _ P. Sadayappan, Job Fairness in Non-Preemptive Job Scheduling, Parallel Processing, 2004. ICPP 2004. International Conference, Publisher: IEEE, ISSN: 0190-3918, ISBN: 0-7695-2197-5

[13] Gaochao Xu, Junjie Pang, and Xiaodong Fu, A Load Balancing Model Based on Cloud Partitioning for the Public Cloud, IEEE Transaction on Cloud Computing,Volume:18,Issue:1,Issue Date:Feb. 2013.

[14] WIKIPEDIA, “Cloud Computing”, http://en.wikipedia.org / wiki /Cloud computing, May 2008.

[15] Brian T. Luke: “K-Means Clustering” http://fconyx.ncifcrf.gov /~lukeb /kmeans.html.

[16] http://paper.researchbib.com/view/paper/18581