# Improving the Quality of service in Cloud Environment

**Venkateshwaran L[1] & Dr.N.Krishnaraj[2]**

[1]Research Scholar, Shri Venkateshwara University, Gajraula, Uttar Pradesh 244236, India

[2]Professor, Department of Computer Science and Engineering, Sasi Institute of Technology and Engineering, Tadeplalligudem 534101, Andrapradesh, India

**Abstract**

One of the special features of the Cloud system is that it permits a user to utilize its services with the help of internet from any location. Cloud services do not restrict the users to depend on fixed locations, a user is independent to utilize the services from any location and this is considered as a special advantage of the cloud system. As these services are independent in nature they do possess various challenges as there are no fixed channels for streamlining the information's. As far as the communication nodes in a cloud environment are considered it is seen that the nodes are not fixed at particular points, these nodes are seen to move from one point to the other constantly based on the user's requirements. As these nodes are not fixed, challenges arise from the fact in maintaining the quality of service rendered by the system. Various factors such as the throughput, load and latency are found to adversely affect the quality of service of the concerned cloud environment. The above stated features are associated with the load balancing strategies with reference to the user requests. This article elaborates certain new load balancing strategies for improvising the quality of services of the cloud environment. A specialized $T^3C$ (Throughput- Traffic-Time-Completeness) strategy has been proposed and in order to accomplish the expected load balancing strategies. Specialized parameters are considered for computing the values of the individual services; such computed values are essentially utilized in the virtual machine selection process for establishing the required load balancing procedures. Such computed values are usually derived from the previously stored log. The methodology thus proposed in this work is found to enhance both the performance and the quality of service parameters of the concerned cloud environment.

Keywords:

Throughput, latency, traffic time, load balancing.

## 1. Introduction

The enhancement and growth of the information technology sector has influenced the growth of the cloud environment. Numerous volumes of resources have been stored in the cloud environment, hence it is considered as a perfect reservoir. Accessing and usage of such resources from the cloud environment can be essentially done through the internet. Cloud environment urges the need for appropriate registration by the concerned users for accessing the stored resources. Any company working on job computational tasks can rely on the cloud services as this is found to be economically in terms of preventing the companies from purchasing a huge number of super computers. Cloud services are found to alleviate such difficulties.

User registration has been made mandatory for accessing the resources of the cloud environment as it consists of a pool of various expensive resources, accessing can be permitted even through appropriate payments. Such registrations simply insist the authenticity of the concerned user. Various types of services have been incorporated by the cloud system for accessing the stored resources. Services are usually made accessible by the inbuilt processors, for which a huge number of the same are essentially deployed. These processors can be in turn accessed only by the appropriate services that are allotted for the processors individually. The cloud environment is seen to offer a wide variety of services namely the IaaS (Infrastructure as a Service), the PaaS (Platform as a Service) and the SaaS (Software as a Service). The above stated services can be easily accessed via internet and various other service providers such as that of the CSP (Cloud Service Provider). It is important to understand that these services may be running at different locations. Numerous virtual machines are found to be deployed in the cloud environment; it is on these virtual machines the service providers would be essentially functioning on. Each of these virtual machines renders specific types of services. Information's obtained in the form of requests are usually handled by these virtual machines. A service provider is restricted to handle only a limited number of requests, anything beyond that number would be allotted to another machine. As the cloud appears to be a massive environment the number of requests arriving from the users would be numerous in numbers, these requests may be for the same tasks at times, hence it becomes important for the cloud services to regulate such requests in terms of numbers and orders. Such regulations would make the accessing approaches easier for the concerned users. Assurance is provided to the users in the processing zones by regulating the

requests appropriately. This assurance comes in the form of a balanced environment, which can be essentially achieved by regulating the processing's between the virtual machines and their services respectively. SaaS type of service stands to be the most predominant form of service in terms of usage as this focuses in the improvements of the cloud platform. Load balancing in this type of service is incorporated by mutually sharing the load progressing towards the cloud controller that is between the various virtual machines, here sharing is accomplished and proceeded on the basis of the available traffic. In this method a separate study process is usually accomplished before the beginning of the sharing procedure in order to understand the capacities of the individual virtual machines. Specialized adoption of the web services are usually accomplished by the cloud system for accessing the services. The time duration allotted to the users for the purpose of requesting and accessing information's is expected to be shorted in order to alleviate the problems of long waiting periods and long queues. The above mentioned problems can be suitably alleviated by minimizing the time durations with suitable load balancing strategies. Such incorporated load balancing strategies are found to enhance the quality of the cloud services. A specialized form of $T^3C$ (Throughput, Traffic, Time and Completeness) based load balancing strategy has been introduced for the purpose of enhancing the attributes of the parameters such as that of the traffic, time complexity, throughput and completeness of the services provided by the various service providers.

## 2. Related Works:

Bernardetta Addis [1] proposed a few load balancing strategies among the widely available approaches in the following chapter. One such is the Honey Bees Inspired Optimization Method, which describes an optimization algorithm known as the Bees Algorithm, that has been inspired from the natural foraging behavior of honey bees, in order to arrive at an optimal solution. The algorithm exhibits an exploitative neighborhood search that is found to be integrated with a random explorative search. This article describes the natural foraging behavior of honey bees, the basic Bees Algorithm and its improvised versions are further explained and thus incorporated in order to optimize several benchmark functions, finally the obtained results are suitably compared with the previously obtained values from the various optimization algorithms.

The second strategy is a Genetic Algorithm (GA) that is found to be purely dependent on the Load Balancing Strategies for Cloud Computing introduced by Kaiyue Wu[2], this approach introduces a novel load balancing mechanism with reference to the Genetic Algorithm (GA). The noticeable feature of this algorithm is that strives in achieving a balance of the load pertaining to the cloud infrastructure while at the same time puts in various efforts in minimizing the make span of a given tasks set. The GA load balancing strategy is suitably simulated by the Cloud Analyst simulator. The next approach is the Load Balancing strategy in Cloud Computing that is found to be dependent on the Stochastic Hill Climbing mechanism – This is viewed as a Soft Computing Approach proposed by Cui Lin [3], as its ultimate aim is to achieve load balancing by means of adopting the various available soft computing strategies. The prime function here is to allocate the incoming jobs to the various available servers by making use of a local optimization approach so called as the Stochastic Hill climbing. The next strategy is the Cross-Breed Job Scheduling that has been adopted for the purpose of reducing the Server Load Using the RBAC at Cloud shown by Kaur, N. Bansal [4], the combination of FCFS and priority has given rise to the Cross Breed Job allocation methodology which is suitably monitored by RBAC (Role based access control). RBAC involves itself in identifying the verifying of the corresponding user's accessibility to a particular content. If RBAC finds a mismatch in the user's accessibility rights then the corresponding user may be denied from using the service immediately and the server would thereafter be minimized.

The next approach appears to be a model for load balancing by Partitioning Public Cloud proposed by Dakshayini [5], that is found to offer a better load balancing model for the public cloud with reference to the cloud partitioning concept. A switch machine is suitably incorporated in this method where the concerned strategies are selected on the basis of the situations. The public cloud is used as the base in this model, where the task of balancing the load on the cloud is performed by segregating the same into various partitions and strategies. It is found that a model that incorporates a main controller, balancers and servers is proposed . Jobs are selected at random and each job is allotted with an appropriate balancer, this allocation process would be perfectly handled by the main controller. Servers holding minimum volumes of load would be appropriately selected by the balancer. It has been observed that the above mentioned strategy efficiently distributes the load among the various available servers by means of identifying the least loaded server and would thereby achieve a balanced cloud system.

A Load Balancing Model that is based on the Cloud Partitioning approach for the Public Cloud Fan Zhang [6], describes a better load balance model for the Job Seekers Web Portal with reference to the cloud partitioning concept where the allotment of the jobs and their corresponding partitions are performed on the basis of the arrival date, in tthe is process the Main Controller (Admin) is utilized for the load balancing tasks. Another approach called as the Enhanced Load Balancing Approach is found to alleviate the Deadlocks in the Cloud tested by Rekha [7], as a result of which a load balancing algorithm has been proposed for neglecting the deadlocks among the various available Virtual Machines (VMs) while the requests transmitted from the users are suitably processed by the VM migration scheme. ShalmaliAmbike [8] found to portray the anticipated results together with the incorporation of the introduced algorithm. It is found that in the process of neglecting the deadlocks the number of jobs to be serviced by cloud service provider simultaneously increases and therefore both the working performance and the business of the concerned cloud service provider enhances suitably.

## 3. Proposed Model for $T^3C$ Mechanism:

The introduced Traffic, throughput, time and completeness dependent dynamic load balancing approach comprises of various functional components such as, Preprocessing, $T^3C$ Computation, Dynamic Load balancing and Target Vm Selection.

### 3.1 Pre-processing:

This methodology comprises of various techniques such as that of the reading strategy, this strategy is related to the service access trace and thereby determines the available unique service and the available unique virtual machine. The various available service providers offering a wide variety of services is also determined by this method, further this method involves itself in segregating the logs and thereby identifies the status of the received service request. Partitioning cannot be done as such, hence log partitioning here is performed on the basis of the various time windows. With respect to this partitioning the service access trace segregation into various groups is thus accomplished. The various other left over parameters are thus selected and computed by the identified and grouped logs. After the completion of the segregation process, these service access traces would be suitably appended to the previously obtained traces or to the appropriate preprocessed log lists. This is essentially incorporated for the purpose of computing the $T^3C$ measure in the next stage of

load balancing.

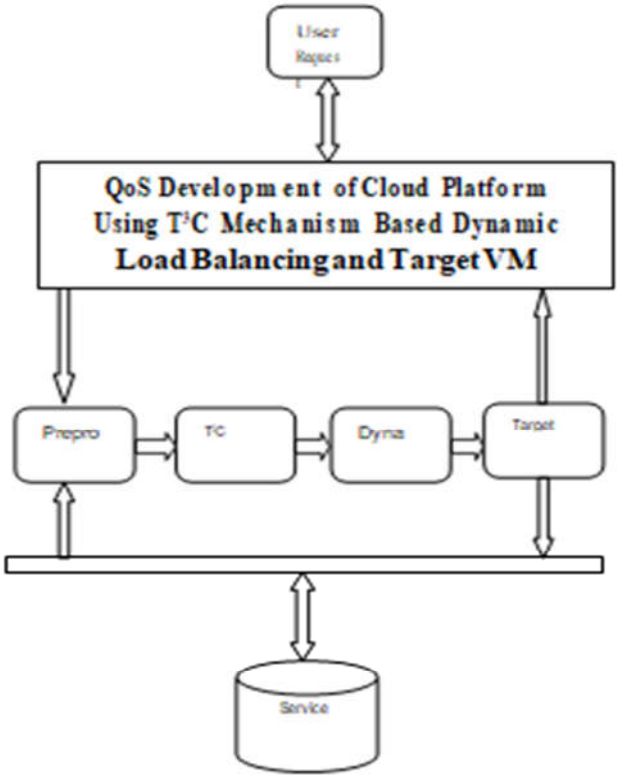


**Fig. 3.1:** Proposed System Architecture for $T^3C$ Mechanism

### 3.2 $T^3C$ Computation:

The proposed method has been found to determine the following three measures namely throughput, traffic rate, time complexity and completeness of each of the individual services that are suitably identified in the previous stages with the help of the preprocessed log. The preprocessed log is suitably utilized in this method and at each time window a unique service is thus identified, the above mentioned method essentially computes the above mentioned measures with the help of the existing log. Load balancing and VM selections are essentially performed in the next stage by making use of the computed measures.

$$\sum T3C + \{Thratio, Trate, Tc, Cratio\}$$

The above discussed algorithm has been found to determine the $T^3C$ measure for each of the individual services that are prevalent at each of the time windows separately and thereby aggregates the same to the concerned set. Load balancing is then performed by making use of the computed values.

**3.3 Dynamic Load Balancing:**

The traffic rate in the Load balancing process is computed on the basis of the number incoming service requests. The initial two stage processes of the method would be computed during the reception of a service request, followed by which the result of the $T^3C$ measure would be efficiently utilized by the method to computes the possible load that could be efficiently handled by the concerned VM which would then be precisely reserved for the execution of the corresponding service. Once the load handling weight corresponding to the individual VMs are determined, their corresponding traffic rates would be suitably determined, followed by which the VM selection mechanism would then be initiated so as to allocate the service request to a specific VM in the cloud. It is found that the above discussed algorithm suitably incorporates the load balancing strategies in the cloud environment with the help of all the functioning modules that have been illustrated so far and then would suitably choose an efficient and optimal virtual machine with reference to the $T^3C$ measure and the VM selection approach.

**3.4 Virtual Machine Selection:**

The load handling weight computed by the dynamic load balancing algorithm is essential made use in the selection of the virtual machines. The computed load handling weight and the service weight list would be utilized in the VM selection approach which essentially segregates the prevalent services on the basis of the obtained weight values, followed by which the cumulative completeness measure would then be suitably computed. It is observed that the VM selection algorithm essentially computes the cumulative completeness measure for each of the individual services together with that of the individual running virtual machines. Based on the above derived values the segregation of the VM's would then take place. If both the obtained values are <> then it is obvious that the above mentioned thresholds and their corresponding VM's are suitably selected as the targets in order to fulfill the service request.

## 4. RESULTS AND DISCUSSION

The proposed $T^3C$ Mechanism that is based on the dynamic load balancing and VM selection approach has been successfully implemented and tested for its efficiency in terms of load balancing. It is observed that the proposed approach has produced enhanced results in terms of the quality of service of the concerned cloud environment. The table below represents the various parameters based on which the evaluation of the proposed method in terms of its efficiency has been performed.

**Table 4.1:** Details of Simulation parameters involved in the evaluation of the proposed load balancing algorithm

| Parameter Name | Value |
|---|---|
| Tool Used | Cloud Sim |
| Number of Service Types | 50 |
| Average Services at each type | 5 |
| Number of users | 1000 |

Table 4 .1 represents the details of various simulation parameters which are involved in the evaluation of the efficiency of the proposed load balancing algorithm. The simulation environment is designed with a set comprising of 50 different typesof services and each of them holds a minimum of 5 similar services belonging to the same type. Evaluation of the concerned environment has been done with respect to the 1000 requests received from the users.

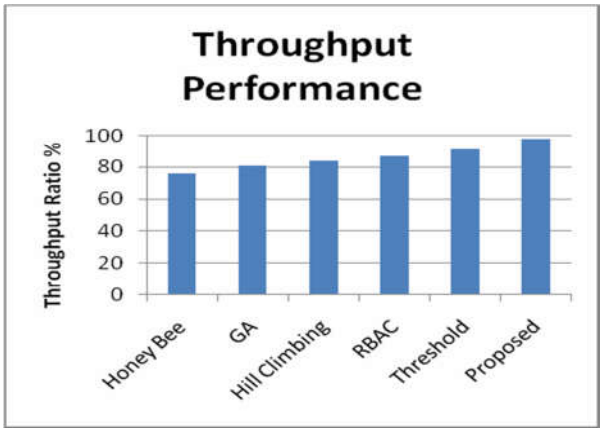


**Fig. 4.1:** Comparison of throughput performance of different methods

The above Fig. 4.1 represents the results of the comparative analysis performed on the throughput performance with various strategies and it clearly portrays that the Efficiency of the proposed approach is better than the other methods.

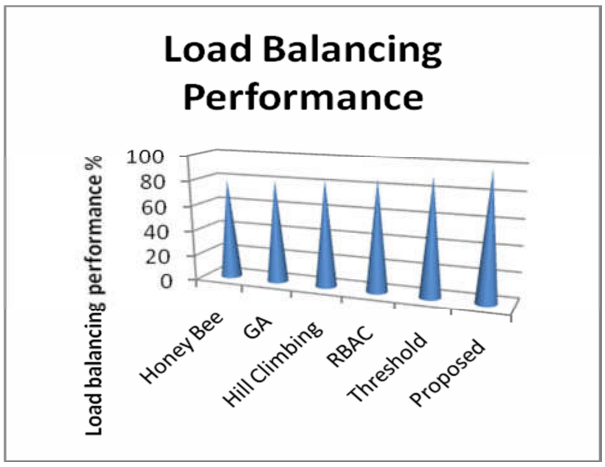


**Fig. 4.2:** Comparison of load balancing performance

Fig. 4.2 represents the comparative result of load balancing performance produced by various methods and it shows that the proposed method's efficiency is superior to the other methods.

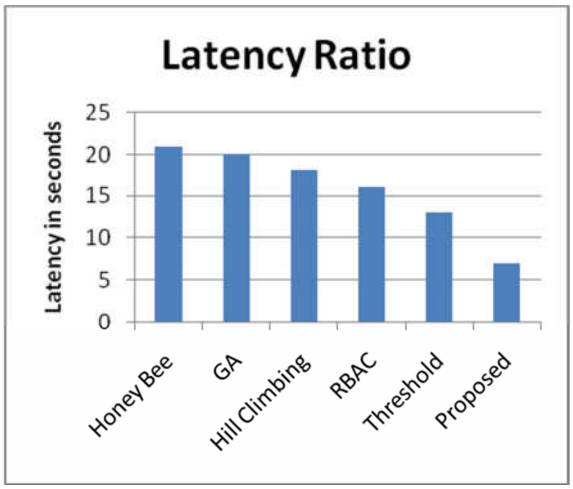


**Fig. 4.3:** Comparison of latency introduced by different methods.

Fig. 4.3 illustrates the comparative result on latency at load balancing proposed by various methods and it shows that the proposed method results in minimized latency when compared with the other approaches.

## 5. CONCLUSION

This research work has proposed a novel $T^3C$ approach that is based on the dynamic load balancing and VM selection approaches for the purpose of enhancing the quality of service of the concerned cloud environment. Similar prevalent services have been identified in the proposed method, which is suitably followed by the segregation of the concerned access traces of the corresponding services into a number of selected time windows. Once the segregation process ceases the evaluation process would commence, evaluation is done on the following parameters, namely, Traffic, Throughput, Time complexity and Completeness measure for each of the services individually at their respective timing windows. With reference to the evaluated measures, the load balancing strategy would essentially compute the load handling weights for each of the individual services. After which the cumulative completeness measure for each of the services corresponding to the individual VMs would be suitably computed with respect to the various computed parameters and measures, which would finally result in the selection of a top valued Vm with respect to the completeness threshold and the current traffic ratio. It is found that the proposed strategy results in an efficient load balancing and enhanced throughput ratio. Further this method reduces the related load balancing latencies that appear to be comparatively higher than the other methods.

References

1. Bernardetta Addis, Danilo Ardagna, Barbara Panicucci, Mark S Squillante& Li Zhang 2013, 'A Hierarchical Approach for the Resource Management of Very Large Cloud Platforms', IEEE Transactions On Dependable And Secure Computing, vol. 10, no. 5, pp. 253-272.

2. Kaiyue Wu, Ping Lu &Zuqing Zhu 2016, 'Distributed Online Scheduling and Routing of Multicast-Oriented Tasks for Profit-Driven Cloud Computing', IEEE Communications Letters, vol. 20, issue 4, pp. 684-687.

3. Cui Lin &Shiyong Lu 2011, 'Scheduling Scientific Workflows Elastically for Cloud Computing', IEEE International Conference on Cloud Computing, pp. 746 – 747.

4. Kaur, M., & Kamboj, S. (2014). Aggressive Migration: An Effective Scheduling Policy. *International Journal of Science and Research*, 146-148.

5. Dakshayini, M &Guruprasad, HS 2011, 'An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment', International Journal of Computer Applications, vol. 32, no. 9.

6. Fan Zhang, Junwei Cao, Kai Hwang, Fellow, Keqin Li & Samee U Khan 2014, 'Adaptive Workflow Scheduling on Cloud Computing Platforms with Iterative Ordinal Optimization', 2014.2350490, IEEE Transactions on Cloud Computing.

7. Rekha, S & Santhosh Kumar, R 2014, 'Priority Based Job Scheduling For Heterogeneous Cloud Environment',International Journal of Computer Science Issues, vol. 11, issue 3, no. 2, pp. 114-119.

8. ShalmaliAmbike, DiptiBhansali, JaeeKshirsagar&JuhiBansiwal 2012, 'An Optimistic Differentiated Job Scheduling System for Cloud Computing', International Journal of Engineering Research and Applications, vol. 2, issue 2, pp. 1212-1214.