Minimize Response Time Using Distance Based Load Balancer Selection Scheme

K. Durga Priyanka
M.Tech CSE Dept.,
Institute of Aeronautical Engineering,
HYD-500043, *Andhra Pradesh, India.*

Dr.N. Chandra Sekhar Reddy
Professor, CSE Dept.,
Institute of Aeronautical Engineering,
HYD-500043, *Andhra Pradesh, India.*

Abstract- Cloud computing is one of the concept in virtual technology. Virtualization means that is not real but its provides the all facilities to all different users. Virtualization provides efficient solution in different number of real time applications. Cloud computing is ready to provide the services efficiently to all users. Different numbers of users are submitting the different jobs. All jobs are store into main controller. Main controllers choose the number of load balancers and provide the solution efficiently without any problem. Dynamically choose the number of load balancers and provide the load balancing solution but it takes more time. Static refresh period based updations are available for execution of jobs. Static refresh period time is the insufficient resources for execution [1] [2] [3].

Reduce the time and increase the scalability with the help of distance based load balancers selection. Main controller chooses the neighbor load balancer and allocate the jobs for execution. Once first load balancer goes to overloaded then again main controller takes decision making to allocate the load in next neighbor load balancer. This is the process continue until load balancing results generation. Analyze the situation with dynamic parameters and perform refresh period operation and provides the sufficient solution for execution. It takes less time for providing the response with reduced time delay [3] [2].

keywords: Cloud partitioning technology, main controller, load balancers, cloud computing, virtualization.

I.INTRODUCTION

Cloud computing use the distributed computing infrastructure and provides the services to all locations of users. Cloud computing provide the attractive services representation process. Those services are infrastructure as a service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS). All large organization offers the cloud services requirement. Load balancing concept is related infrastructure as a service concept [4] [5].

Existing load balancing solution generate with the help of load balancers and main controller. Main controller chooses the load balancers without any distance parameter consideration in implementation. It provides the low scalability, less availability and efficient results. Response time related problems are generated here [3] [4] [5].

The above system problems we overcome create the new load balancing architecture. After arrives the jobs in main controller, main controller choose the distance based nearest load balancer selection here. We got the load balancing solution with in the fewer amounts of time and best availability in our implementation [1] [2].

II.RELATED WORK

Large and small organizations migrate to cloud computing infrastructure or environment for increasing the scalability results to the users. All different locations of user's jobs are reached to cloud. Cloud allocates the jobs in different number of servers for execution. After completion of execution provide the return back results to the users. In this infrastructure at execution environment many issues are generated. Those issues we described below [1].

We suggest the round robin algorithm for load balancing environment. Data center gets the tasks from different location users and directly allocate into virtual machines. Select the virtual machines randomly from total number of virtual machines. All different tasks allocate into circular order of virtual machines. After completion of alignment in first circular order then we start the reallocation of jobs in the same order. This is the repetitive process. It takes more time for completion of load balancing environment creation [1] [2].

New load balancing algorithm was introduced as a extension dynamic round robin algorithm implementation. It provides the advantage with reduced power consumption. Every physical machine has different number of virtual machines. All virtual machines are in running mode, any new virtual machine is come to same physical machine, and this is not accepted. Virtual machine is not wait automatically virtual machine join into another physical machine and save the power levels. After completion of virtual machines all the servers are in shut down mode [2].

Next extended load balancing approach was introduced that is called active monitoring load balancing technique with the help of first come first serve basis. It follows the allocation and de-allocation procedures. After we got the tasks from different number of users equally we divide the tasks allocate into different number of virtual machines. Any new requests are coming from different number of users, those things also de-allocate into another new virtual machine. Previous all the virtual machines are in overloaded state then we call the new virtual machine for control the load [3] [4].

New Load balancing solution was previously introduced that is called Combination of Round robin and first fit algorithm. These two algorithms focus on perfect resource utilization without wastage. Count the number of tasks according to create the virtual and physical machines. All remaining servers are shutdown finally. It provides the power consumption solution [4] [3].

Enhanced Equally distributed load balancing algorithm was introduced. It can provide the solution as a maximum throughput solution. Identify the each and every job size checks the resources in virtual machine. It’s not sufficient then we add the resources and execute the all jobs successfully with the help of distributed load balancing algorithms implementation. All jobs are executes efficiently without any faults. Any faults are occurring those faults we control with the help priority concept environment implementation. All jobs are executed successfully and we provide the efficient solution [1] [2].

New self organized load balancing environment solution we create here. After we got the jobs from different locations users, local servers are sufficient are not we checking at first. Otherwise we allocate the jobs in global server’s specification process. It provides the soft solution for all jobs execution process [2].

Max-min algorithm was introduced for execution of tasks. Identify the tasks categorize the tasks into two subsets. Those subsets are minimum execution time tasks and maximum execution tasks process. Reduce the waiting time first allocate the minimum execution time tasks then after we give the preference to large tasks [2] [1].

Next Two phase load balancing technique divides the tasks and allocate into parallel number of processors. First categorize the tasks into min execution time tasks and maximum execution time tasks. Two different categories of tasks allocate into two different numbers of schedulers. These two schedulers are executing the tasks and provide the response within the time [3] [2].

III. EXISTING SYSTEM

Finally inelastic cloud provides the services as a pay per use manner. Load balancing problems are generated because of static servers. Here present in static servers there is no predictability property. Another load balancing architecture is created with the help of main controller and load balancer. Identify the number of tasks provide the load balancing solution. Choose the load balancers [1] [2] [3] from different geographical locations without any distance consideration.

IV. DRAWBACKS

It shows the performance and scalability is low.

The above all proposals provide the different load balancing results with different performance environments. These all proposals environment is highly expensive.

V.PROBLEM STATEMENT

First conceptual framework follows the cloud partitioning technology for providing the load balancing solutions. After implementation of cloud partitioning technology also some new issues are generated here. Those issues are related performance and efficiency. Different number of geographical locations has different number of load balancers within the cloud. Consider the tasks select the load balancer randomly without any distance consideration. Some load balancers are available in so far distances. It takes more time for providing the load balancing solution. Sometimes it does not provide load balancing solutions within the time. When to extend the time we have no awareness. Here there is no reasonable solution [2] [3].

Now here we develop the new load balancing strategy with distance dimension. After completion of cloud partition process, different geographical locations have different load balancers. Now here we allocate the distance for each and every load balancer. Count the number of tasks or jobs chooses the load balancer based on distance. We provide the load balancing solution with in less amount of time. Here now we set the reasonable refresh period without delay. It gives the better performance and efficiency load balancing solution compare to all previous proposals [3] [1].

VI.PROPOSED SYSTEM

Cloud computing have many number of nodes. Nodes are partitioned into number of clusters. Partitioning nodes of clusters are generated based on cloud clustering technique [1] [3]. Our proposed model consists of main controller and distance specified load balancers. Main controller controls all different distances load balancers efficiently with reasonable refresh period amount of time. Every load balancer controls the load in different number of internal nodes. Previous cloud partitioning techniques consist of static parameters. Now it’s changed as a dynamic parameters or reasonable parameter [2][1].

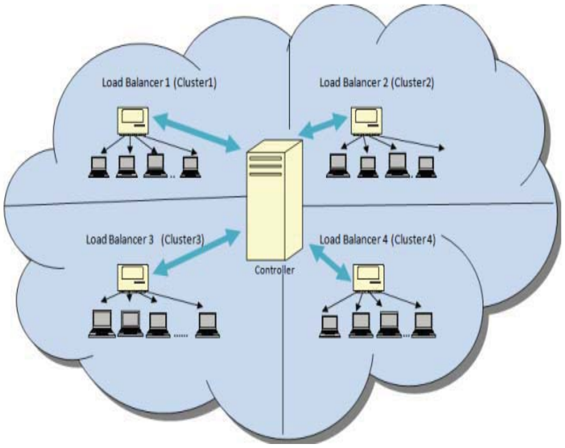
Load Balancer 1 (Cluster1)
Load Balancer 2 (Cluster2)
Load Balancer 3 (Cluster3)
Load Balancer 4 (Cluster4)
Controller

Fig1: PROPOSED SYSTEM ARCHITECTURE

Different locations of user’s jobs arrive to main controller. Main controller is available in cloud environment. Count the jobs or tasks choose the suitable cluster based on distance. Once cluster nodes status is overloaded, then send the information to cloud controller. Cloud controller choose the another neighbor cluster nodes. Continue the process until we get the load balancing choose the neighbor load balancer. We got the load balancing solution with in the less amount time. It takes less response time and save the resources. This is solution to increase the scalability environment process. Static parameters are showing the problems like more amount of delay because of insufficient time. Now in this paper dynamic parameters was introduced here based on requirements it’s possible to increase the time interval for execution with reduced. time delay solutions in implementation [1] [2].

VII. ALGORITHM

Distance based neighbor load balancer selection algorithm:
Input: jobs or tasks or instances, load balancers, main controller
Output: load balancing solution with less response time
Actions:
Run algorithm
Time to check balance?
Suspend application
IF load is balanced, resume application
Re-partition the load
Distribute data structures among processors
Resume execution

VIII. EXPERIMENTAL EVOLUTION

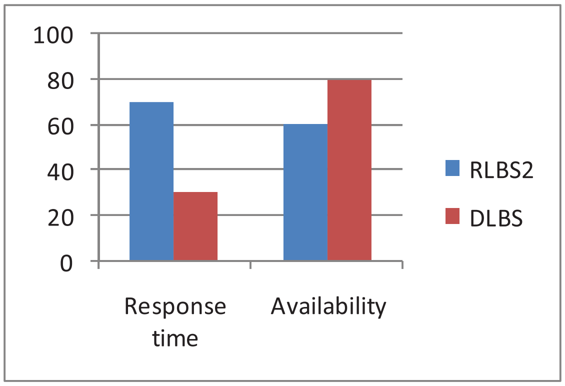

Fig2: RLBS and DLBS

Fig2 contains the performance graph results related random load balance selection (RLBS) and Distance based load balancer selection. Here we calculate the response for existing and proposed system. Proposed system takes the less amount of time for providing the response. Static parameters are not providing the sufficient results. Dynamic parameters have more number resources. Proposed system provides the more amount of availability of resources content in implementation. Actual output graph is as follow:

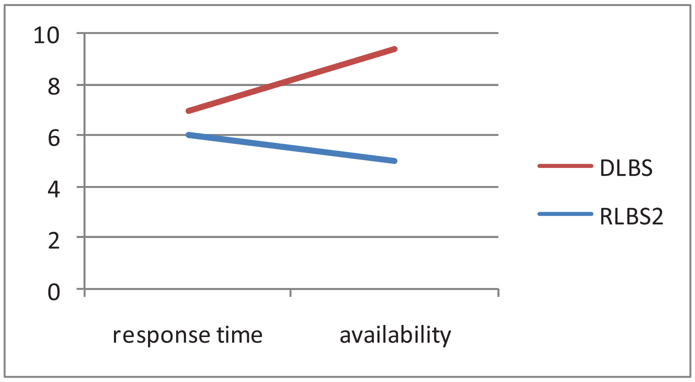

Fig3: DLBS and RLBS

IX.CONCLUSION AND FUTURE WORK

Existing cloud partitioning framework provides the better solutions compare to previous approaches. In cloud partitioning also some issues are available here those are scalability and performance issues. Selection of load balancer without any distance consideration it shows the problems. It takes more time for providing the best response results content. Now in this paper we show the experiment to provide the load balancing solution with in the amount of response time. This solution is generates based distance load balancer selection only. It's here perform the refresh period based on the requirement.In future develop the load balancing strategies every time provide the guaranteed load balancing solutions.

X. FUTURE WORK:

Some of the tasks are missing without execution in dead line. We develop the new frameworks for execution of all tasks within the dead line.

REFERENCES

[1] Gaochao Xu, Junjie Pang, and Xiaodong Fu, A Load Balancing Model Based on Cloud Partitioning for the Public Cloud, 2013

[2] M.Aruna, D.Bhanu, R.Punithagowri A Survey on Load Balancing Algorithms in Cloud Environment, 2013

[3] Soumya Ranjan Jena, Zulfikhar Ahmad Response Time Minimization of Different Load Balancing Algorithms in Cloud Computing Environment, 2013

[4] Yatendra Sahu, R.K. Pateriya Cloud Computing Overview with Load Balancing Techniques, 2013

[5] Benjamin Busjaeger, Jonathan Chu, Daniel Ormond, Self-Partitioning Cloud Datastore for Scalable Transaction Processing Proposal, 2012

[6] Miss.Rudra Koteswaramma ,Client-Side Load Balancing and Resource Monitoring in Cloud, 2012

[7] Zuhair Khayyatz Karim Awaraz Amani Alonaziz Hani Jamjoomy Dan Williamsy Panos Kalnisz, Mizan: A System for Dynamic Load Balancing in Large-scale Graph Processing, 2013

[8] T.Sakthisri1, S.Pallavi2BALANCING BLOCKS FOR DISTRIBUTED FILE SYSTEM IN CLOUDS BY USING LOAD REBALANCING ALGORITHM, 2013

[9] Ram Prasad padhy,p gutham Prasad rao,LOAD BALANCING IN CLOUD COMPUTING SYSTEMS,2011