Dynamic Load Balancing In Public Cloud Using KP Model

[1]N Maheshwar Reddy, [2]S.Krishnaveni
[1]M.TechStudent, [2]Assistant Professor
[1,2]Dept of Software Engineering, SRM University, Kattangulathur,Chennai,India-603203
[1]reddy.mahesh36@gmail.com, [2]krishnaveni.s@ktr.srmuniv.ac.in

***Abstract-* Cloud computing is the use of computing resources that are delivered as a service over a network. Cloud computing can be used for in different ways; in present market it is the attractive one which can be use by everyone. Load balancing is the cost effective concept in cloud computing environment and it shows influence on the performance. An optimum load balancing can make cloud computing more cost effective and it gives fulfillment results for user. Load balancing will perform in public cloud based on the cloud partitioning concept and here we have to choose different methods for different problems. In existing system, Round-Robin algorithm used for ideal status and game theory can be used for normal status of the system and as well as for load balancing process. When number of clients send request to cloud at that time we need to face traffic at this time we cannot schedule task to the partitions to avoid this traffic here we are using KP (Koutsoupias, Papadimitriou) model. The nature of the KP model, can be use for the reduce compilation of the tasks which we got from user due to that we can schedule task to the partition without facing traffic.**

Keywords –load balancing model; public cloud; cloud partition; game theory; KP model

I. INTRODUCTION

Cloud computing [2] is the delivery of computing resources to different users from the different locations and it is a information pool and by using this we can get data what we need according to our requirements. The name of cloud computing set based on the architecture of the cloud and in this we have lot of systems which has the inter connection between them. Cloud computing to give into the care of other services with a user's data, software and computation over a network. Load balancing [3,7] is a computer networking process for sharing data or information whatever based on the requirement across different computing resources, such as computers, network links. Load balancing main aims to makes best resource use, maximize throughput, minimize response time, and avoid overload of any one of the resources. By using multiple systems with load balancing instead of a single system it may increase reliability through redundancy. In this cloud computing, we have four types: public, community, private and hybrid.

II. RELATED WORK

Load balancing [3,7] is a computer networking, when client communicate with cloud for any resources the services provider provides data from the storage area then it sends to the clients in their required format. We can say load balancing is the balancing loads between the clients and server or cloud. The load balancing in the cloud, when client search any data from the network it will be connect with database and search what we need then it gives acknowledgment to client and this load balancing concept performed in the public cloud can be done after cloud partition. Dynamic load balancing [6] in the public can be vary in the sense every time we can't predict the client task complexity and computational time because some tasks have less complexity and few may not .Load balancing in the cloud is still a new problem that needs new architectures to adjust the role that the load balancing play vital role in improving performance and managing stability. When we see this system model, in this load balancing process can be done after cloud partitioning and the cloud partitioning used to manage large cloud. After cloud partition load balancing will start before that we have to know cloud partition in public cloud can be happen by the geographic locations. Generally in cloud, we have main controller and balancers in each partition and this load balancing problem solved by the main controller and balancers. When client send task to the cloud automatically it approach to the main controller and at that time it will decide which partition has to get task after assigning task to balancers it decides how to assign task to nodes. In each partition we have one balancer when main controller get job it assigns to the cloud partition and then communicates with the balancers in each partition to refresh information about status. According to cloud partition, when task arrives to main controller it choose different partition in different situations because here cloud partition can be divided into three i.e. ideal, normal and overloaded and this status according to load degree(virtual machines).

In Existing system, we use round robin algorithm to schedule the task to the idle processors. R-R algorithm [4] is the simplest one in load balancing algorithms, which new request to the server in the queue. The algorithm doesn't has the status information but in public cloud environment every will not perform the same process this concept will not match to some nodes. The second algorithm, Game theory is used to schedule the task to the normal processors. Two algorithms were used to perform task scheduling. Thus it takes more computation time to schedule the task. In this we saw about ideal and normal status, if we see the overloaded means node is not available and it doesn't get any tasks until it become to normal. The advantages of this game theory[4] is : Game theory gives insight into less known aspect which arises in situations of conflicting interests. Game theory develops a framework for analyzing decision making in such situation where interdependence of firms is considered. At least in two-person zero-sum games, game theory outlines a scientific quantitative technique that can be used by players to arrive at a optimal strategy. when we see disadvantages for this concept The assumption that players have the knowledge about their own pay-offs and pay-offs of others is not practical. The technique of solving games involving mixed strategies particularly in case of large pay-off matrix is very complicated. All the competitive problems cannot be analyzed with the help of game theory. When number of tasks arrives to cloud at that time we have to face traffic problem due to reason we can't distribute the loads to clients. To prevent this situation here we are proposing the KP model for avoiding traffic i.e., by using this method we reduce the compiling time of all tasks which we got from the clients.

III. PROPOSED METHODOLOGY

In cloud computing environment, one of the main components of a distributed system is the distributed process scheduler that manages the resources of the system. The efficient usage of the large computing Capacity of a distributed system depends on the success of its resource management system. A distributed process scheduler manages the resources of the whole system efficiently by distributing the load among the processors to maximize the overall system performance. To improve the job response time, proposed system uses the job scheduling strategy. Where the client sends the number of task to the main controller, it assigns the job based on the load status of balancers. Balancers are nothing but geographically distributed with the number of servers. The proposed architecture is given below:

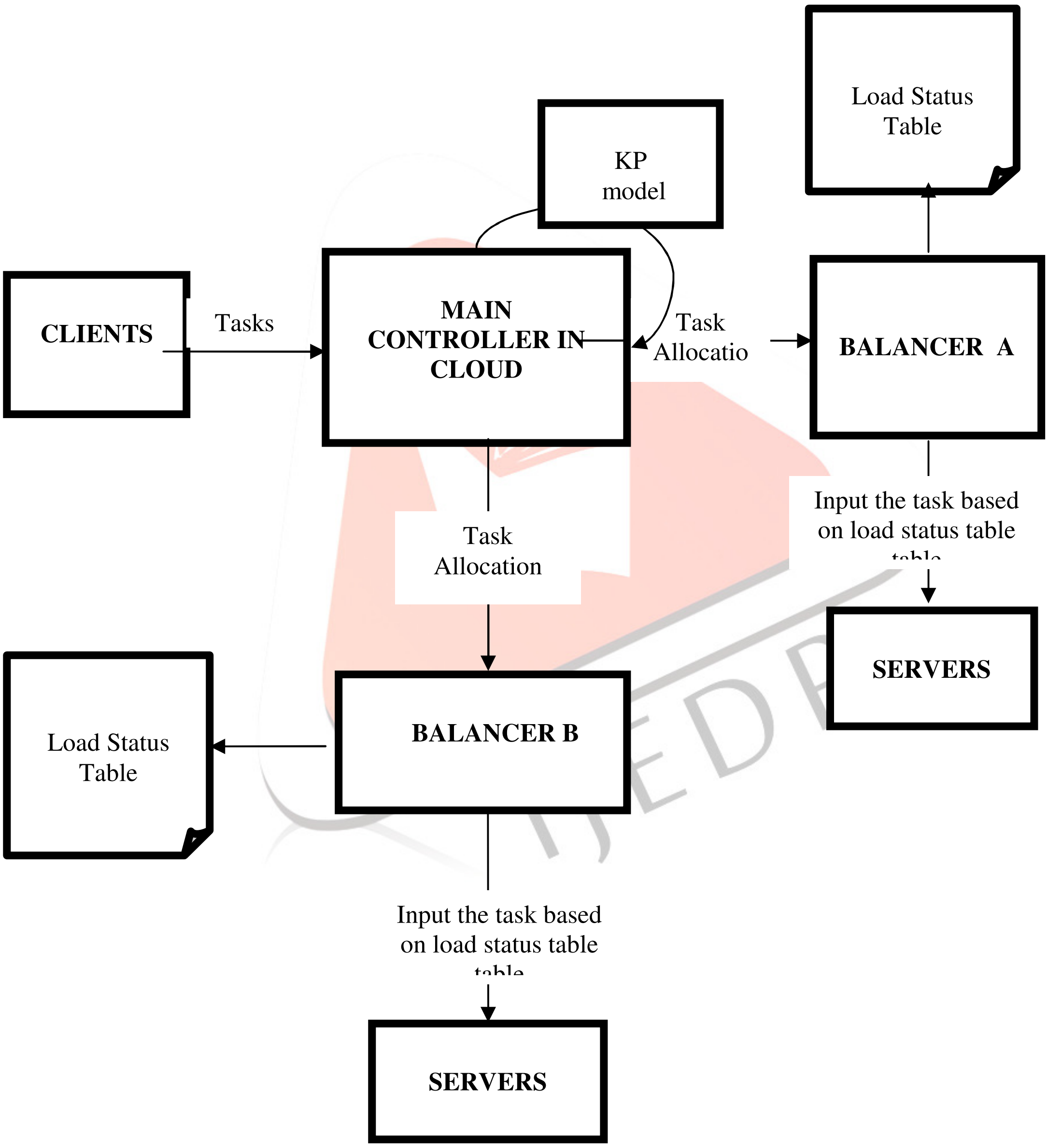

Fig 1 proposed system model

When number of clients send task to cloud we have to face the traffic congestion to prevent this problem here we are using KP model. Through this method we can reduce the compile time of every task which we got from the different clients. In load balancing we have algorithms, round-robin, ant colony algorithm and equally spread current execution algorithm by using these concepts we can perform fast process but we can't reduce the compile time according to these algorithms. When we apply this concept in main controller environment, it will assign task to balancers after that it sends to nodes. The KP model algorithm can be given below:

Step 1: set of tasks from clients

Step 2: Pi^j the probability of task i to go on machine (balancer) j
Step 3: The expected cost of task i, if it decides to go on machine (balancer) j with $Pi^j =1$
Step 4: $Ci^j = 1i + \sum_{K=i} S\ pk^j+lk$

Where,

li is speed of processor

$\sum_{K=I} S\ pk^j$ it intimates the probability of the task

Lk length of the task can be classified by task complexity.

C indicates compile time for each task

Step 5: In a Nash equilibrium, i assigns non zero probabilities only to the machines that minimize Ci^j.

Example: Consider 3 tasks lengths are 2, 4, 1 and 2 and one virtual machine, its processor speed is 2.20 GHz. First task of probability is to go on machine is 1/3 and same like second task is go on machine is1/3 and remain last task is go on machine is 1/3.

When we see the compilation of task one is on machine

$$C1^1= 2.20*10^9+ 1/3*2 = 1.6\text{(understanding purpose consider 1.6 value)}$$

Then compilation of task 2 on machine

$$C2^1= 2.20*10^9 + 1/3*4 = 2.3\text{(for example)}$$

Same like task 3 on machine

$$C3^1= 2.20*10^9+ 1/3*1 = 1.3\text{(for example consider 1.3 is compile time value)}$$

If we see the compilation time of task 1we can get value is nearly 1.6(for example) and for task2 value is 2.3(for example) and the task 3 compilation values is 1.3(for example). According to this compilation values we can say task 3 has less compilation value than task 1 and 2 so that task 3 can be go on machine because it has less compilation time value. Likewise by using this KP model we can reduce traffic congestion by scheduling tasks which has less compile time.

The KP model is used here for scheduling tasks by reducing the compilation time as well as we can be reduce traffic congestion also before that all this process will performed by using cloud simulation tool. Through this tool we can create cloud environment within our requirements. In this simulation process we need to set main controller, balancers along with what we need for this work. CloudSim [8] a new, generalized, and extensible simulation framework that allows seamless modeling, simulation, and experimentation of emerging Cloud computing infrastructures and application services. By using CloudSim, researchers and industry-based developers can test the performance of a newly developed application service in a controlled and easy to set-up environment. Based on the evaluation results reported by CloudSim, they can further fine-tune the service performance. The main advantages of using CloudSim for initial performance testing include: time effectiveness it requires very less effort and time to implement Cloud-based application provisioning test environment and flexibility and applicability: developers can model and test the performance of their application services in heterogeneous Cloud environments with little programming and deployment effort.

CloudSim[8] offers the following novel features: (i) support for modeling and simulation of large-scale Cloud computing environments, including data centers, on a single physical computing node; (ii) a self-contained platform for modeling Clouds, service brokers, provisioning, and allocation policies; (iii) support for simulation of network connections among the simulated system elements; and (iv) facility for simulation of federated Cloud environment that inter-networks resources from both private and public domains, a feature critical for research studies related to Cloud-Bursts and automatic application scaling. Some of the unique features of CloudSim are: (i) availability of a virtualization engine that aids in the creation and management of multiple, independent, and co-hosted virtualized services on a data center node and (ii) flexibility to switch between space-shared and time-shared allocation of processing cores to virtualized services. These compelling features of CloudSim would speed up the development of new application provisioning algorithms. It's a perfect method for creating the cloud environment through this we can implement operations which will going to perform in next stage to make successful project and this is a free resources in present situation any one can use in their required format in which way they want while getting perfect output.

IV. APPLICATIONS

Medical applications- The proposed work can implement in the medical applications for access the information from the hospital server according to the user query. The hospital server may places in the different geographical areas and each server contains the node systems for processing the different query process. With the proposed method, schedules the jobs for the node systems and process the query accurately.

V. CONCLUSION

We proposed KP model, by using this concept we can reduce the traffic congestion in process of allocating tasks to the balancers. Through this method we can reduce the compile time of the every task which we get in the public cloud environment. We have some specialty is there to use this concept to compare all load balancing algorithms. In load balancing concept we have round-robin, ant colony algorithm and equally spread current execution algorithm for improving speed of processing in the sense getting quick solution of clients request but all this concepts can't be reduce the compile time of each task. By using this KP model we can schedule task to balancers through preventing traffic congestion and main advantage of this procedure is it reduce the each task of the compilation. Good load balancing criteria will improve the performance of the whole cloud environment and in this area we do not have common method to adjust for all different situations. Every method has different benefits in particular area not in all strategies or situations.

REFERENCES

[1] Chhabra, G. Singh, Qualitative Parametric Comparison of Load Balancing Algorithms in Distributed Computing Environment,14th International Conference on Advanced Computing and Communication, July 2006 IEEE, pp 58 – 61.

[2] A. Rouse, Public cloud, http://searchcloudcomputing.techtarget.com/definition/public-cloud, 2012.

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

[4] M.Randles, D. Lamb, and A. Taleb-Bendiab , A comparative study into distributed load balancing algorithms for cloud computing, in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications, Perth, Australia, 2010, pp. 551-556.

[5] Madhooshi M. 2007. Developing an integrated model for calculating the customer lifetime value, The 4th International Management conference, Iran.

[6] Wenhong Tian, Yong Zhao, Yuanliang Zhong, Minxian Xu, Chen Jing(2011)**,**A dynamic and integrated loadbalancing scheduling algorithm for cloud datacenters**,** University of Electronic Science and Technology.

[7] Ram Prasad Padhy (107CS046), P Goutam Prasad Rao (107CS039).”Load balancing in cloud computing system” Department of Computer Science and Engineering National Institute of Technology, Rourkela Rourkela-769 008, Orissa, India May, 2011.

[8] http://www.cloudbus.org/cloudsim.