Load Balancing Technique in Public Cloud Environments using Combination of Heuristic Function and KNN Classification

Rahul Dongarde M.E. Scholar Branch: CSE TIEIT, Bhopal (M.P.) Devashri Deoskar Asst. Professor Guide Dept: CSE TIEIT, Bhopal (M.P.) Amit Saxena Head of Department Dept: CSE TIEIT, Bhopal (M.P.)

ABSTRACT

The proper utilization of virtual machine in cloud computing environments, manage the process of load. The process of load balancing increases the efficiency and utility of cloud environments. The major components for the balancing of load is allocation of resource and distribution of tasks. For the allocation and distribution of tasks various swarm-based algorithms is proposed and enhanced the performance of cloud environments. In this paper proposed the virtual machine classification technique based on the distribution of tasks using heuristic function. The proposed algorithm is combination of KNN algorithm and ant colony optimization. The ant colony optimization is meta-heuristic function used for the process of optimization. The proposed algorithm simulated in CLOUDSIM 3.0 and measure the performance of proposed algorithm.

Keywords

Cloud Computing, Load Balancing, Heuristic Function, KNN, ACO.

1. INTRODUCTION

The proper management of virtual machine in cloud computing environments, increase the performance of cloud network. The cloud-based services depend on the process of resource allocation and execution of task[1]. The execution of task depends on the virtual machine. the unmanaged process of virtual machine creates the scenario of balancing problem in cloud environments and decrease the performance of cloud networks. In current scenario various authors and researcher develop the methods and process for the proper selection of virtual machine for the allocation of resource and task. The utilization of cloud networks in every filed now a day, so the responsibility of cloud network is increase[2-4]. In the process of responsibility, the management of load is big issue. For the managements of load used various load balancing algorithms. The load balancing algorithm manage the load in terms of static load balancing and dynamic load balancing. The static load balancing process used the traditional CPU scheduling algorithm for the allocation of resource and management of task. Instead of that the dynamic load balancing technique used swarm-based techniques such as PSO, ACO and many more algorithms[9]. At exhibit, the field of load balancing in cloud server farms is a hot issue. To accomplish the heap balancing of cloud server farms, it is important to pick the ideal virtual machine effectively during the time spent conveying errands. Most existing work has concentrated on the issue of how to accomplish the prompt load balancing inside a calculation cycle in which the proposed approaches can get the ideal assignment arrangement strategies for the current [5], [6]. There exists a confinement of excessively

focusing on the ideal load balancing approach for the present sending issue and, in this way, to make the effectiveness diminished and clients holding up time delayed pointlessly. As is outstanding burden balancing is perceived as a method for giving agreeable administration execution to clients while boosting the accessibility and use of the entire cloud framework instead of the last reason[10]. The categorization of virtual machine based on load gives the better distribution of load according to the selection policy. In this paper used KNN classification technique for the selection and grouping of virtual machines. the grouped virtual machine procced the load for the allocation of resource atask[8]. The rest of paper discuss as section II describe the process of KNN and ACO, in section III discuss the proposed algorithm. in section IV discuss the simulation of proposed algorithms and finally discuss conclusion and future work in section V.

2. KNN AND ACO

In this section describe the process of KNN and ACO. The KNN is classification algorithms and ACO is swarm-based optimization algorithms. Here the KNN algorithms distribute the load of virtual machine according to the selection of ant colony optimization[12]. The KNN classification algorithms works based on single attribute of virtual machine the maximum capacity of machine and load the job for the allocation of resource. The process of ACO and theprocedure of virtual machine enhancement in cloud conditions, virtual machine space of insect province streamlining[11]. The mapping of heap of virtual machine credit as per their fake ants required some standard determination and parameter. In view of parameter gauge the virtual machine closeness of two distinct burdens. What's more, those virtual machines are most comparative goes through the procedure of arrangement and increment the limit of virtual machine[12].

Let VMS is a virtual machine set and N is the aggregate counterfeit ants and plausibility of insect choice is s_1, s_2, \ldots . .sn, now discover the choice probability of two ants in given arrangement is

Where si and sj is the divergent likelihood of two distinct ants. Presently appraise the estimation of appetence of ants is

Where αi and βi is ants whose determination probability is greatest as far as another ants the proportion of choice of ants is characterized as 100/N

In view of determination probability appraise the estimation of counterfeit wonder esteem

$$\Delta \tau i = \frac{A.si}{ACP(i+j)}\dots\dots\dots\dots\dots(3)$$

Where An is consistent marvel esteem Presently every emphasis of pheromone esteem is augmentation and decrement as indicated by their choice likelihood. The inference of all-inclusive appetence likelihood is

$$p_{ii}^{k} = \begin{cases} [\tau_{ij(t)}] \\ 0 & otherwise \end{cases} \alpha. [k_{ij}]\beta & if \ j \in j \dots \dots \dots (4) \end{cases}$$

Where kij gives the data of heuristic hunt space and measure the determination plausibility of fake ants

Lastly getting the ideal virtual machine of KNN characterization.

3. PROPOSED ALGORITHM

The process of algorithm is combination of the KNN, ACO and virtual machine. The process of virtual machine optimization reduces uncertain capacity of load. The optimal value of virtual machine passes through the KNN classifier. the KNN mapped the relation with load of cloud network. The process of descriptions given below. Mapped the virtual machine load pattern $P_r \in \Re^D$ in the KNN. The mapped load of KNN define the group level of machine. The grouping of class to load creates maximum capacity of virtual machine.

- 1. Input: VM1, VM2,.....Mn load patterns Output: optimal allocation of load
- 2. Compute $VM_{(P k)}$ and k similar(RD)
- 3. for all load $\in maxM_{(p_t,k)}$ do
- 4. estimate $-Relation(p_t, load)$
- 5. end for
- 6. $KNN \leftarrow load_{(p_t,k)}$ {label of class map
- 7. for all $VMs \in KNN$ and $load \in max$ do
- 8. iterate vm pattern(load) and rmap pattern(VM, max)
- 9. if $vm_{(class)}$ then
- 10. $KNN \leftarrow KNN \cup \{VM\}$
- 11. end if
- 12. end for
- 13. for all $VM \in KNN$ do
- 14. Update VM(Label) and class({capacity})
- 15. end for
- 16. allocate resource for virtual machine
- 17. return VM

4. PROPOSED MODEL



Figure 1: Block Diagram of Proposed Algorithm for Load Balancing in Public Cloud Network

5. SIMULATION OF PROPOSED MODEL

To interface with different administrations in the cloud and to keep up the assets in an adjusted way to satisfy the necessity of assets/foundation by those administrations, a few procedures are required. In light of a center arrangement of elements in the three basic cloud administrations, for example, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). To assess the execution of distributed computing systems in distributed computing situations for the heap adjust and asset administration, here we are utilizing different quantities of procedures, for example, LB-BC and proposed strategy. For the further execution and correlation for execution assessment we utilized java programming dialects with NetBeans IDE 8.0.1 devices for finish usage/comes about process.



Figure 2: window present that the User Database and Data center in our Cloud Computing Environment implementation

Simulation Complete



Figure 3: window present that the for-User Database, Overall Response Time in Cloud Computing Environment implementation using proposed method

6. RESULT ANALYSIS

Table 1: Shows the response time and processing time analysis using LB-BC Method

TIME	OVERALL RESPONSE TIME	DATA CENTER PROCESSING TIME
AVERAGE	367.214	0.401
MINIMUM	305.891	0.124
MAXIMUM	398.212	0.827
DATASET: UB1-DC1		

Table 2:	Shows the	response	time and	processing	time
	analysis	using Pro	posed M	ethod	

TIME	OVERALL RESPONSE TIME	DATA CENTER PROCESSING TIME
AVERAGE	244.871	0.256
MINIMUM	178.333	0.098
MAXIMUM	280.039	0.454
DATASET: UB1-DC1		

Table 3: Shows the response time and processing time analysis using LB-BC Method

TIME	OVERALL RESPONSE TIME	DATA CENTER PROCESSING TIME
AVERAGE	377.242	0.606
MINIMUM	340.441	0.381
MAXIMUM	430.219	0.887
DATASET: UB2-DC2		

Table 4: Shows the response time and processing time analysis using Proposed Method

TIME	OVERALL RESPONSE TIME	DATA CENTER PROCESSING TIME
AVERAGE	345.965	0.392
MINIMUM	280.894	0.198
MAXIMUM	360.582	0.696
DATASET: UB2-DC2		

7. PERFORMANCE ANAYSIS



Figure 4: show that the comparative performance of overall response time between LB-BC method and Proposed method using UB1-DC1 dataset with average time, minimum time and maximum time. Here proposed method show that the improved performance compared to the LB-BC method.



Figure 5: show that the comparative performance of data center processing time between LB-BC method and Proposed method using UB1-DC1 dataset with average time, minimum time and maximum time. Here proposed method show that the improved performance compared to the LB-BC method.

Comparative performance of Overall Response Time b/w LB-BC and Proposed Method for UB2-DC2 dataset



Figure 6: show that the comparative performance of overall response time between LB-BC method and Proposed method using UB2-DC2 dataset with average time, minimum time and maximum time. Here proposed method show that the improved performance compared to the LB-BC method.



Comparative performance of Data Center Processing Time b/w LB-BC and Proposed Method for UB2-DC2 dataset

Figure 7: show that the comparative performance of data center processing time between LB-BC method and Proposed method using UB2-DC2 dataset with average time, minimum time and maximum time. Here proposed method show that the improved performance compared to the LB-BC method.

8. CONCLUSION & FUTURE WORK

In this paper proposed KNN based virtual machine load distribution algorithm for load balancing in public cloud environments. The KNN classifier map the load of virtual machine according to the selection of the ant selection of virtual machine. The KNN and ACO work together and distribute the load and allocation of resource according to their process requirements. The process of load selection based on the maximum capacity of virtual machine. The proposed algorithm reduced the load variation in cloud environments and increase the capacity of cloud computing efficiency. Proposed should have the ability to select one virtual machine or more in the set obtained by the KNN approach to deploy tasks and thus to make the benefit acquired by users and cloud data centers maximized at the same time.

9. REFERENCES

- [1] Jia Zhao, Kun Yang, Xiaohui Wei, Yan Ding, Liang Hu, and Gaochao Xu "A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment" IEEE Transactions on Parallel and Distributed Systems, VOL. 27, 2016. Pp 305-316.
- [2] AtyafDhari, Khaldun I. Arif "An Efficient Load Balancing Scheme for Cloud Computing" Indian Journal of Science and Technology, Vol 10, 2014. Pp 1-8.
- [3] C.Antony, C.Chandrasekar "A Hybrid Multi-Threaded Task Scheduling and Knapsack Load Balancing in Multiple Cloud Centers" International Journal Of Engineering And Computer Science ISSN: 2319-7242 Volume 6 Issue 2 Feb. 2017, Page No. 20324-20332.
- [4] Liang Tong, Yong Li and Wei Gao "A Hierarchical Edge Cloud Architecture for Mobile Computing" IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications. Pp 1-9.
- [5] YunanZhai, Ming Hu, and Jia Zhao "An Energy-efficient Task Scheduling Approach based on Firefly Algorithm" Journal of Residuals Science & Technology, Vol. 13, No. 8, 2016, Pp 38-384.

- [6] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and MateiZaharia "Above the Clouds: A Berkeley View of Cloud Computing" 2009. Pp 1-23.
- [7] Saurabh Kumar Garg, Steve Versteeg and Rajkumar Buyya "SMICloud: A Framework for Comparing and Ranking Cloud Services" Pp 1-9.
- [8] Wei-Tek Tsai, Xin Sun, Janaka Balasooriya "Service-Oriented Cloud Computing Architecture" Seventh International Conference on Information Technology, IEEE, 2010. Pp 684-689.
- [9] Saurabh Kumar Garg, Steve Versteeg, Rajkumar Buyya "A framework for ranking of cloud computing services" Future Generation Computer Systems, 2013, 1012–1023.
- [10] Jesus Garcia-Galana, Pablo Trinidada, Omer F. Ranab, Antonio Ruiz-Cortes "Automated Configuration Support for Infrastructure Migration to the Cloud" Preprint submitted to Future Generation Computer Systems, 2015. Pp 1-19.
- [11] Qiang Yang, Wei-Neng Chen, Zhengtao Yu, Tianlong Gu, Yun Li, Huaxiang Zhang and Jun Zhang, "Adaptive Multimodal Continuous Ant Colony Optimization", IEEE, 2017, Pp 191-205.
- [12] Jesus Maillo, Sergio Ramirez, Isaac Triguero and Francisco Herrera "kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors Classifier for Big Data", Elsevier, 2016, Pp 1-38.