IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

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

Vol 5, Issue 5, May 2018

Load Balancing Technique in Cloud Computing Environment

[1]Nakul Kabrawala, [2] V. Vivek
[1] M.Tech Student, [2] Assistant Professor
[1][2] Department of Computer Science & Engineering, School of Engineering & Technology, Jain University, Bengaluru, Karnataka

Abstract: **-- More and more, we are seeing technology moving to the cloud environment. Task scheduling, Load balancing, and resource allocation are significant stages for any cloud environment. Scheduling is the process of mapping and managing task or processes into the available resources. So while allocating the resource in cloud environment distribution of virtual machines and other resources should equally load to achieve the better performance. In our paper we present a detailed survey on existing resource allocations and proposed a novel framework to allocate resources and balance the load using binary tree and heap property keeping different cases in considerations.**

Index Terms: - Cloud Computing, Resource Allocation, Task scheduling, Heap property, VM ware, Service Provider.

I. INTRODUCTION

The cloud computing is became a more and more important concept for now a days in modern computing. In other words if we are going to consize the definition, then it is a delivery of computing services in terms of server, storage, database, networking, software and analytics over a common layer called internet or cloud [2]. Now a day's most of the companies are going to have the same layer of environment because by implementing such cloud environment we get the following benefits, such as

- Fast computing,
- Low cost per bit of data storage as well as processing,
- High reliability in terms of backup and recovery of data in span of time,
- Ease of handling of data, and
- Better resource utilization as on demand.

Out of the above benefits of the cloud, the most of the companies are mainly adopt cloud due to because of its better resource allocation and utilization as on demand [1, 4]. If we are looking for the allocation of resource, then the cloud provides a better provision for the resource allocation during the processing of task. In resource allocation we are mainly going for the scheduling process of task which is the most important aspect for any processing of data or modern computing [3]. In traditional mode of computing, the scheduling approach was not so good, so each time the user has to be interacted with the system to load the process, which indirectly increases the processing time and slow down the execution [6, 2]. Hence we are mostly interested towards the cloud architecture where we are having the better resource allocation strategy and scheduling mechanism. Cloud provides a great platform for the day to day computing terminology which accelerates the computing process. Cloud architecture basically consists of its unique mechanism through which it can help us to provide the fastest computing [5, 8]. It also provides the distributed computing approach, which is really a transformative approach for scheduling of task that totally changes how registering administrations are created, valued and conveyed [1]. It also allows to permits us to get the privileges values for a user so that the user can able to achieve the fastest computing. In its arrangement aspect (which is also known as cloud load balancing), it confines the task in terms of their fragment basis and then used to generalize the problem statement in to own segment [2]. Whenever a task is submitted for processing, it allows managing application or workload demands by allocating resources among multiple computers, networks or servers. To make it feasible, it first computes the total approximate time execution of task and then depending on the availability of resources, it assigns the task to achieve the fastest processing. To accomplish such process we need to segmentise the task in to cluster pattern so that the ultimate backup will be attained and we can get the concept of batch processing [3]. In such process the user needn't worry to enter the task always, and also get a complete freedom regarding the scheduling burden. If we look for programming aspect, then it utilizes the Hive or Pig over the platform of spark to schedule the task.

II. RELATED WORKS

As we know that, the cloud computing is a modern trend of computing in which almost all relevant organizations are using for their business. It doesn’t matter whether they are using for storage, launching of websites, or retrieval of data. We also know that, the cloud computing is also used in distributed computing environment, where the progression of cloud assignment and their planning depends on virtual machine. A virtual machine is used to identify the resource availability by identifying the problem statement and used to provide the service as per the resource availability [1, 2]. During the process of relocation of task, the VM ware is uses its scheduling mechanism to identify the resource and then it allocate the job to the corresponding resource. In the VM ware the major component which is responsible for such activity is Virtual Manager which itself get operated on the basis of Virtual tree being obtained from the Resource tree and the collaboration of VM scheduler. In the process of allocation of the task, we use five allocation strategies [11, 7]. They are the arbitrary designation procedure, the full arrangement distribution technique, the succession portion methodology that the cloud errands are arranged by the execution time before the undertakings are relegated to the virtual machines by turns, the grouping allotment system that the virtual machines are arranged by the execution speed before the errands are allocated to the virtual machines by turns and the covetous system that the heap adjusting is considered [8]. In its construction point of view, the cloud basically works on three tire concept which is a Model-view-controller based approach. Here the view layer is the user oriented layer (browser) form which the user will send the task for processing as well as view the outcome [13, 11]. The Model is the database where the data are stored permanently and the controller is the service provider, which interact with the view as well as the model as per the requirement and the instruction provided by the Virtual Manager present in the System. Here the service provider may uses any of the said application like SaaS, PaaS and IaaS as per the resource availability and the problem statement [14, 4]. But the most important problem get created when the task is integrated. At this case the generation of the respective parse tree is difficult and so the VM ware cannot able to decide the optimal solution [11, 12]. Hence the service provider will be in ambiguity and the system gets halted.

III. PROBLEM DEFINITION

In traditional days programming, the major problem arises when the task is subjected under the processing. In such cases the task has to be reassembled by the system which takes lot of time as well as makes the algorithm to be more and more complex. It also increases the users burden, as the user has to interact with the system each time. It also uses the concept of System-level scheduling which is a meta-task execution, sub-optimal solution of which is given by heuristics to speed up the process of finding a good enough answer [11]. If we look to Real time scenario then the problem is quite optimized and well structure as compare to the present situation. Here we are going to implement the concept of VM ware machine which is an ultimate prototype of SAAS model where a user can achieve the real time application [7, 9]. In other scenario the task has to be signifies the process structure [12]. As we know that the major role played in VM ware is the manager, who will allocate the resource as per the demand of execution by the CPU and thus the major burden of the programmer is get reduces.

IV. PROPOSED SYSTEM

Figure-1 shows the typical generalized cloud architecture. In the traditional concept the scheduling of task is systematically and sequentially. So the major drawback of the system is time complexity. But in the generalized cloud computing the problem has overcame due to the VMware which manages the scheduling of job using the batch processing technique where the user has to load the task once and then he needn’t have to interact with the system any more in between the execution. In generalized form, the cloud structure has three main service applications such as, SAAS, PAAS and IAAS. The role of these service providers is to provide the service on demand and instruction given by the virtual manager present in the VM ware. Whenever users try to achieve the service they simply use the browser and give the task by its URL. The task so submitted by the user is get recognize by the cloud service provider with the help of Virtual tree over the internet. The service provider then analyzes the form of request and accordingly distributes the job among the resources. But when the task contains the schema free structure data, then identifying of cluster of data is bit difficult for the service provider. Hence we are trying to modify the system by adding a small component called as “Presentation Service” layer.

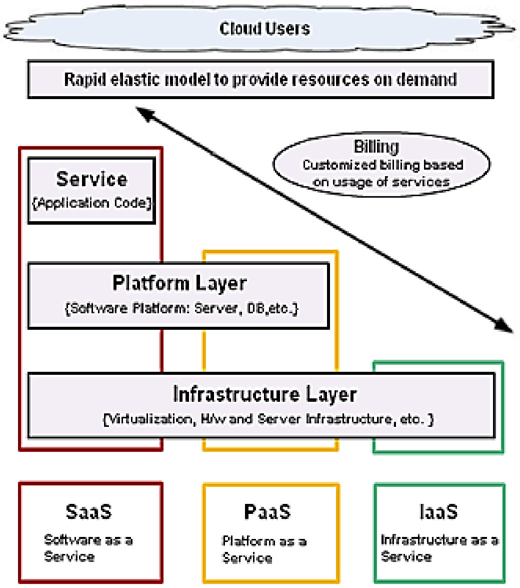

Cloud Users
Rapid elastic model to provide resources on demand
Billing
Customized billing based on usage of services
Service
{Application Code}
Platform Layer
{Software Platform: Server, DB,etc.}
Infrastructure Layer
{Virtualization, H/w and Server Infrastructure, etc. }
SaaS
Software as a Service
PaaS
Platform as a Service
IaaS
Infrastructure as a Service

Figure-1 Generalized cloud architecture

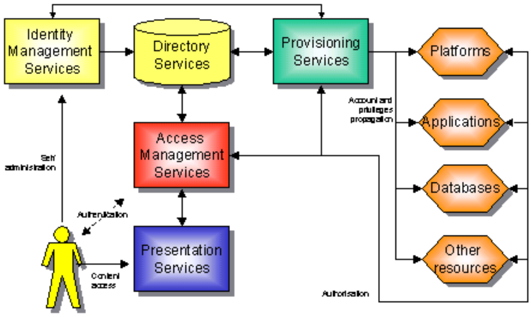


Figure-2 Cloud as per our proposed system

Figure-2 Cloud as per our proposed system Thus by implementing this we can identify the schema of the data and can make the cluster code easily. The clustering of data can be then operated on the command of VM ware to schedule the task. The VM ware uses the concept of tree mechanism to identify the cluster code and then rearrange on the basis of heapify to manage the heap of the tree. So once the task is gets processed over, then rearrangement of the data code is easier and faster, which in turn reduces the time and the space complexities.

V. VM-SCHEDULER PROCESS FLOW

In general, to explain the process of VM scheduler is a typical one but we can able to explain on the basis of tree formation concept, which we are mostly used in data structure. Let us consider a binary tree having the three nodes a H1, H2, H3.

Let Host Tr= {H1,H2,…,Hn} which can be conceive as physical machines organized as a tree, which are available to process VM requests generated from user through the task. For a typical time interval VM requests are gathered at VMM (Virtual machine manager) and are defined as VMTr= {VM1,VM2,…,VMn}. The process of plotting VMTr□HostTr.

It should be done in such a way that, it will improve the resource utilization and reduce response time.

Now to get the above things done, we use here the response data from VM ware according to the following flow control mechanism as showed in Figure-2.

$$\frac{RQVM}{AVM} = \begin{cases} 1 & Perfact\,VM \\ (0,1) & Possible\,VM \\ (1,+\infty) & Reject \end{cases}$$

Where, RQVM -> Request Queue for the VM ware.

AVM is the Advance Management of VM ware for the Request Queue. In general the ratios of these are used for calculating the performance of the system. From the above equation, if the ratio is 1(exactly) then the service request queue performance is better and hence the cloud can perform better. On the other hand if the ratio factor is lies in between 0 to 1, then the performance is bit slower and it is called as cloud overloading effect. Similarly, if the value of ratio factor is more than 1, then it is called cloud halting. In operation point of view, RQVM is requested by VM ware when a user use the browser to submit the task. AVM is Available VM at cloud service provider side. When the data is made available to the cloud through the browser layer, it is first identified by the host machine. The host machine generates the corresponding tree with the help of VM host tree controller. The tree so far obtained is then subjected to the cloud service provider for the further resource allocation. The schematic diagram is as showed below.

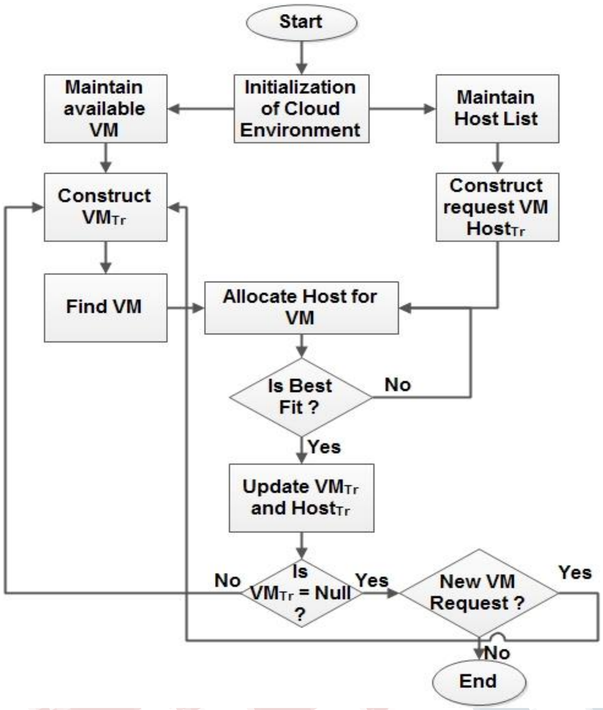
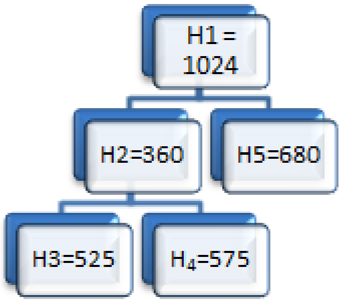
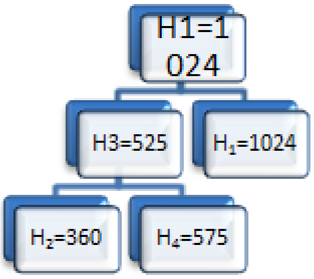


Figure 2 Flowchart of our proposal to identify the cluster code

When a virtual machine request arrives at the panel for finalization of the task result at the scheduler, then the scheduler uses the tree approach to calculate the final result. Here the approach may in the form of generalized binary tree which is a heap balancing [10]. As we know that in Heap balancing the root of the tree should be larger than its left most derivation and should be less than its rightmost derivation. Hence in this way it tries to summaries the result and heapify the overall task record. Thus by doing so, it recognizes all the individual component of the problem and then integrates at the final stage to conclude the result [10,11]. This saves a space as well as time complexities for a problem instruction. To explain the mechanism in better manner, let us consider there are five host nodes H1 to H5 present at the data center having memory specification (in MB) 1024, 360, 525, 575, 680 respectively. Now when these data set is send to the cloud for processing, the service provider first looks for its VM tree. As the data is send serially so it reaches to host in the form of following manner as showed in diagram. Now by looking to the diagram, we know that the tree so formed by the host is not balance and hence the heap can't be done. So as per the sequence the data set send to the VM host allocator for generation of best heap. Now if we look to balancing then the tree is evaluated from the left most derivation as in following manner.





Now by doing so the tree is balance from its left span but still it is remains unbalanced with respect to root. So again it is send VM host allocator for further balancing and then the simplified tree obtained as showed below, which is perfectly balance and heapified.

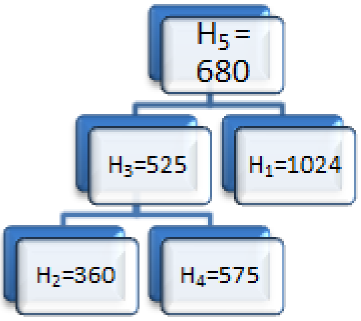


Figure- 3 Priority Heap structure of the tree

In this above example priority heap tree is formed according to its RAM configuration of available VM. Meanwhile consumers request a VM having memory specification 480MB arrives at VM scheduler. Now VM scheduler would calculate RQVM/AVM at the root node (i.e. at level H5). Now as the heap tree is already formed the search will continue in left child tree. This will end at H3 where the value is found to be 0:91. Next VM host is H2 and calculated value is 1:33 which is greater than 1. Then it's the real time scenario of the VMware machine which recognizes the schedule by admiring its service protocol over the span. Here the virtual manager tries to reconfiguring the overall task by keeping its history and provides the feedback to the provider so that the final integration of the result should be accurate and concrete. So the system can able to analyze the problem by itself and make the user free from burden.

VI. CONCLUSION

From the above, now we can came to know that no doubt, the cloud computing is best option for the modern task computing, but the performance of the computing needs the proper load balancing and the resource allocation. If we look to the generalized form of the cloud then, as per our study, the computation may get slow down when the RQVM and AVM factor is more than 1. Such cases are usually get arises when the scheduling process by the VM ware is not proper. So we present a novel technique to handle load balancing issues through simple heap based tree type data structure (formed by VM ware) which are easy and faster way in realizations. We achieve this approach by implementing a new layer called “Presentation Layer” in the traditional cloud computing mechanism, so that the raw data so arrived for processing must be interpreted in such a manner so that, the VM ware used to generate the tree for the respective data set having the RQVM and AVM ratio factor as in between (0,1) or 1, and used to allocate the proper resource for processing. On the other hand we also get a better option for the data backup through the cluster heads which also reduces the risk factor for the data loss during processing and makes the user comfortable.

REFERENCES

[1] J. Barbosa and B. Moreira, “Dynamic job scheduling on heterogeneous clusters,” in 8th International Symposium on Parallel and Distributed Computing, ISPDC 2009, 2009, pp. 3–10.

[2] T. A. Syed, S. Musa, A. Rahman, and S. Jan, “Towards Secure Instance Migration in the Cloud,” in proceedings of International Conference on Cloud Computing (ICCC), 2015, pp. 1–6.

[3] D. Nurmi et al., “The Eucalyptus open-source cloud-computing system,” in Nineth IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID’09), 2009, pp. 124–131.

[4] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008.

[5] I. M. Abbadi and A. Ruan, “Towards Trustworthy Resource Scheduling in Clouds,” IEEE Trans. Inf. Forensics Secur., vol. 8, no. 6, pp. 973–984, Jun. 2013.

[6] D. S. Milojičić, F. Douglis, Y. Paindaveine, R. Wheeler, and S. Zhou, “Process migration,” ACM Comput. Surv., vol. 32, no. 3, pp. 241–299, 2000.

[7] P. Mell and T. Grance, “The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technolog,” Natl. Inst. Stand. Technol., 2011.

[8] X. Xu, H. Hu, N. Hu, and W. Ying, “Cloud task and virtual machine allocation strategy in cloud computing environment,” in Communications in Computer and Information Science, 2012, vol. 345, pp. 113–120.

[9] S. Sotiriadis, N. Bessis, Y. Huang, P. Sant, and C. Maple, “Towards decentralized grid agent models for continuous resource discovery of interoperable grid Virtual Organisations,” in 2010 5th International Conference on Digital Information Management, ICDIM 2010, 2010, pp. 530–535.

[10] S. Sotiriadis, N. Bessis, Y. Huang, P. Sant, and C. Maple, “Defining minimum requirements of inter-collaborated nodes by measuring the weight of node interactions,” in CISIS 2010 - The 4th International Conference on Complex, Intelligent and Software Intensive Systems, 2010, pp. 291–298.

[11] S. Sotiriadis, N. Bessis, and N. Antonopoulos, “Towards inter-cloud schedulers: A survey of meta-scheduling approaches,” in Proceedings - 2011 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2011, 2011, pp. 59–66.

[12] B. Xu, C. Zhao, E. Hu, and B. Hu, “Job scheduling algorithm based on Berger model in cloud environment,” Adv. Eng. Softw., vol. 42, no. 7, pp. 419–425, 2011.

[13] S. Ghanbari and M. Othman, “A priority based job scheduling algorithm in cloud computing,” in Procedia Engineering, 2012, vol. 50, pp. 778–785.

[14] E. I. Djebbar and G. Belalem, “Tasks Scheduling and Resource Allocation for High Data Management in Scientific Cloud Computing Environment,” in International Conference on Mobile, Secure and Programmable Networking, MSPN, 2016, vol. 10026 LNCS, pp. 16–27.