Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 12, December 2014, pg.521 – 528

RESEARCH ARTICLE

Study of Deadline Sensitive Resource Allocation Scheduling Policy in Cloud Computing

Sasmita Parida, Suvendu Chandan Nayak
Department of Computer Science and Engineering
C.V.Raman College of Engineering, India
sasmitamohanty5@gmail.com, suvendu2006@gmail.com

Abstract- Now a day's Cloud computing is a most attractive computing model of internet technology, which provides resources on-demand as per use and pay basics. The most challenging process is allocation and reallocation of resources according to the unpredictable demands of the user simultaneously improving the performance of computing. Different researcher has proposed different resource allocation algorithm for resource allocation in cloud computing. During resource allocation different parameters are considered. In this paper we discussed different scheduling algorithm. Among which we studied the deadline sensitive leases scheduling algorithm. Haizea is a popular resource manager that supports four kinds of resource allocation policies: immediate, best effort, advanced reservation and deadline sensitive. The deadline lease based scheduling algorithm accepts more number of leases. It is possible only by dividing a lease into multiple slots and backfilled into the already accommodated leases.
Keywords- Resource Allocation, Scheduling Algorithm, Lease, Deadline, Swapping, Backfilling

I. INTRODUCTION

A new paradigm which provides Internet-based services, by providing highly scalable distributed computing plat-forms is called cloud computing. Where computational resources are offered as a service as per the user request. But consumers do not need to understand how it all computes and can easily access various services via the Internet. Mainly cloud computing is the collection of datacenters, where each data center consists of number of physical machines. Each physical machine has its own resources with fixed amount like CPU, Main memory, storage area, I/O and bandwidth. When a request is send by the user to the datacenter. The virtual machine (VM) will be created according to the request at the physical machine.

The main challenge is creation of number of virtual machine in a physical machine to process the user request. The user is unknown about the physical machine. The user request which is called lease consists of different parameters like CPU Time, Main memory, Storage area. To schedule resources different models and algorithms are proposed. Basically the resource allocation is done in terms of leases. Haizea lease manager [1] is most popular in cloud computing. Four types of leases are supported by Haizea they are as Immediate, Best effort, Advance reservation and Deadline sensitive.

Most of the time it is too difficult for cloud providers to successfully execute all the users' requests which come to them in a instant due to lack of resource or scheduling of resource. As reported in [2] resource allocation model is categories into three ways.

- Effort with a focus on Data Center processing Resources.
- Efforts with a focus on Data center Network Resources.
- Efforts with a focus on Energy Efficient Data Center Resource Allocation.

In [1] the authors specified a better way to support deadline sensitive leases in Haizea by trying to minimizing the total number of leases rejected by it. The proposed work of dynamic planning based scheduling algorithm is implemented in Haizea which can admit new leases and prepare the schedule whenever a new lease can be accommodated. The experiments results show that it maximizes resource utilization and acceptance of leases compared to the existing algorithm of Haizea. In this paper the deadline lease based scheduling algorithm is studied in details along with backfilling algorithm.

The layout of the remaining of this paper is given as follows: In Section II, a review of literature has been presented in the area of resource allocation in cloud computing. The detail study of deadline based lease scheduling algorithm along with swapping and backfilling policy is presented in Section III. The study is summarized and the future works are illustrated in Section IV.

II-RELATED WORKS

When a user makes a request to run an application in the cloud .The application goes to the data center where a physical machine handles the request and allocates a or more than one Virtual machine to run the application. So the resource allocation scheduling is required for proper utilization of resources. Resource scheduling is one of the most important problems in cloud computing. However, it is a big challenge for efficient resource scheduling algorithm design and implementation (since general scheduling problem is NP complete) [3].

A net work aware resource allocation in distributed clouds is proposed [4].The authors minimized the maximum distance, or latency, between the selected data centers. Where the user has large task and it needs more than one virtual machine. Haizea also provides a better advanced reservation and deadline sensitive type of leases. Capacity leasing in cloud systems using the open nebula engine is discussed in [5]. In [6] more on resource leasing and the art of suspending virtual machines is briefly discussed. Sotomayor also discussed combining batch execution and leasing using virtual machines [7].

Except Haizea different types of resource scheduling algorithms are proposed. Concentrating on specific subsystems of Large Scale Distributed Systems, such as [8] on the performance of memory systems, or only deal with one or two specific SLA parameters. Petrucci et al. [9] or Bichler et al. [10] discussed one general resource constraint and also Khanna et al. [11] only focuses on response time and throughput during resource scheduling.

Haizea is an open-source VM-based lease management architecture [12].It provides the following architecture:

- Haizea is a resource manager: depending on who you ask, a "resource scheduler".
- Haizea uses leases: lease is some form of contract where one party agrees to provide a set of resources to another party.
- Haizea is VM-based: Haizea's scheduling algorithms are geared towards managing virtual machines, factoring in all the extra operations (and overhead) involved in managing VMs.
- Haizea is an open source.

The above architecture supports the following types of lease where the lease is nothing but the user request to process the application. The leases required a virtual machine or a group of machine to compute the application.

- Advance reservation leases, which must start at a specific time.
- Immediate leases, which must start right now, or not at all.
- Best-effort leases, which will wait in a queue until resources become available.
- Deadline sensitive: Cloud user can ask for specific resources. This lease has specific start Time, end Time (deadline), duration. If cloud provider can provide resources in this time duration then the lease is accepted else it is rejected.

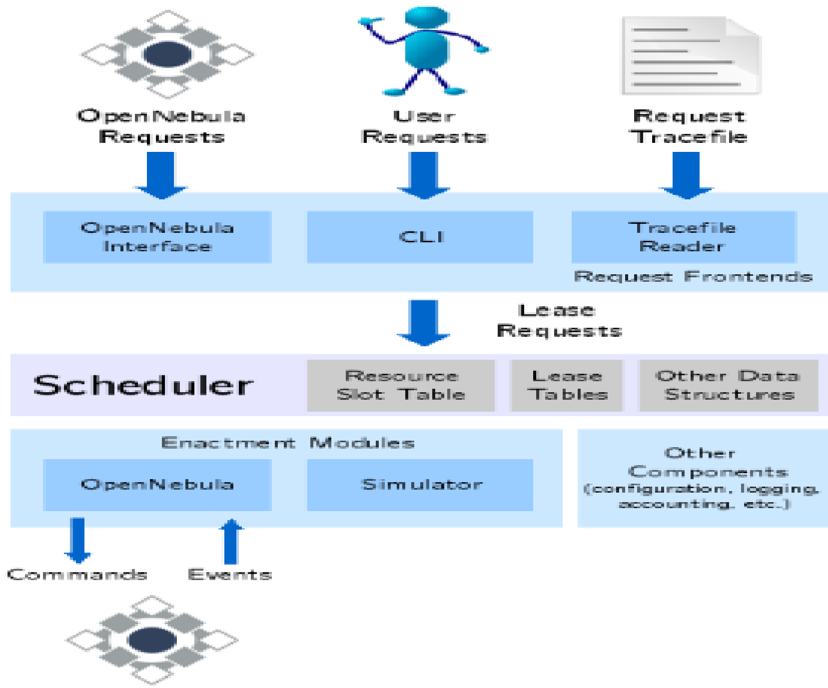

Figure 1: The Haizea architecture

The Haizea architecture is shown in Figure 1 the requests are in the form of lease. Once a lease is accepted, it cannot be rejected. The scheduler plays a vital role while trying to accommodate newly arrived leases. The scheduler has to verify that whether it is already accommodated leases are still able to run successfully or not. Haizea uses simple resource allocation policies for deadline sensitive leases. It basically tries to find out a single slot that is required in between start Time and end Time of the given lease. By which it be can allocate the requested resources for the user. If such a time slot it is not found by the scheduler then that lease will be rejected.

Resource allocation algorithms that handle a variety of usage scenarios are needed. Typically, user requests may arrive and leave at any time. A user may also make requests for more resources as time progresses. User may need specific amount of resources for the application in cloud which can be given for some specific amount of time. So for every resource a start time and given end time (deadline) can be considered. This type of requirement should be given to Haizea in the form of deadline sensitive lease. When a request with deadline sensitive lease comes to Haizea, first it tries to find out a single slot which can provide the required resources for required duration of time. If Haizea can find such a time slot, it accepts the newly arrived deadline sensitive lease. If it cannot find such a single slot then Haizea reschedules already accommodated deadline sensitive and apply best effort leases. For this, first it finds out which leases can be rescheduled and then how they can be rescheduled. Leases having start time or end time greater than or equal to the start time of newly arrived lease are considered for being rescheduled. Once these leases are fixed, Haizea adds the newly arrived deadline sensitive lease to the list [13].

In this paper deadline sensitive scheduling algorithm with best effort is studied. Along with the swapping and backfilling algorithm is discussed which increases the performance of deadline sensitive algorithm in further section.

III-STUDY OF DEADLINE SENSITIVE

Actually a lease parameters are number of nodes (number of virtual machine), physical resource for each node like CPU utilization and Main Memory, start time, duration and end time which is called deadline. A Lease object is maintained in Haizea to stores lease's information after receiving the lease. Then the scheduler tries to schedule the lease.

Haizea has several scheduling options that specify how Haizea selects resources and schedules leases. There is no guarantee that a submitted best-effort lease will get resources to complete the processing within a certain time limit. If the system is flooded with lots of advance reservation and immediate leases then best effort leases will not have enough resources to run on. The consumers of best-effort leases may not like to wait so long to get resources [1]. They will start submitting their requests as advance reservation leases rather than best-effort leases, to be assured that the submitted requests will be completed within a certain time limit. As a result, the system will have more advance reservation leases which will increase fragmentation of free resources. As there are less best-effort leases, very few of these fragmented unused resources will be utilized by them.

Thus, the system utilization will go down. To handle this kind of situation, deadlines can be associated with best-effort leases for those consumers who want to get resources for their best-effort leases within certain time limit. These kinds of leases can be considered as deadline sensitive leases [1].

Basically Deadline sensitive leases are considered to be preemptable but a limitation is there. It is preemptable only if the scheduling algorithm of Haizea can assure that it can be completed before its deadline. So it will assure the consumers that their request will be completed within a certain time limit. There are several other options one can modify in the scheduling section, such as what backfilling algorithm and swapping is to be used.Let us consider the example [1] for clear study of deadline sensitive scheduling.

Table 1: Information of Leases.

Lease no.	Nodes	Submit time (AM)	Start Time (PM)	Duration	Deadline (PM)
1	2	11.10	12.00	40	01.20
2	3	11.20	12.00	20	01.20
3	4	11.30	12.00	20	01.20
4	1	11.40	12.30	40	01.20
5	2	11.50	12.40	10	01.20

As per Table-1 if allocation will be accrued then lease 4 and 5 could not be scheduled. Before scheduling they cannot meet their deadline as shown in the Figure-2 where 4 numbers of nodes are used as resources.

Time → Node ↓	12.00-12.20	12.20-12.40	12.40-1.00	1.00-1.20
d				3
c			2	3
b	1	1	2	3
a	1	1	2	3

Figure 2: Simple allocation policy

Allocation using Swapping policy

To avoid this type of problem swapping policy can be used. We can swap two consecutive leases if and only if, the first lease required fewer amounts of resources than the second lease. And after swapping they must meet their deadline.

Time → Node ↓	12.00-12.20	12.20-12.30	12.30-12.40	12.40-1.00	1.00-1.10	1.10-1.20
d	3		4	4	4	5
c	3	2	2			5
b	3	2	2	1	1	1
a	3	2	2	1	1	1

Figure 3: Allocation using Swapping policy.

Figure 2 demonstrates that 1st lease 3 will be allocated because it needs 4 nodes then lease 2 where the requirement of node is 3. Now there rest 3 lease (1, 4 and 5) where lease 4 needs only one node but lease 4 and 5 needs 2 nodes. Swapping is applied to allocate lease 4 then lease 1 to meet its deadline and lease 5 at last.

Table 2: Information of Leases.

Lease no.	Nodes	Submit time (AM)	Start Time (PM)	Duration	Deadline (PM)
1	2	11.10	12.00	20	12.30
2	3	11.20	12.00	40	01.00
3	2	11.30	12.00	50	01.50
4	4	11.40	01.00	20	01.50

Let us consider the Table 2 for allocation to understand the problem occurred by using swapping policy. For which we have to consider the policy which is called Backfilling.

Time → Node ↓	12.00-12.20	12.20-12.40	12.40-1.00	1.00-1.20	1.20-1.40	1.40-1.50
d						
c		2	2			
b	1	2	2	3	3	3
a	1	2	2	3	3	3

Figure 4: Simple allocation policy

Figure 4 shows the allocation of the leases using simple allocation policy where lease 4 and 5 are not allocated. Let us try to do it by swapping policy shown in the figure 5 below.

Time → Node ↓	12.00-2.20	12.20-2.40	12.40-1.00	1.00-1.20	1.20-1.40	1.40-1.50
d						
c	2	2				
b	2	2	3	3	3	
a	2	2	3	3	3	

Figure 5: Allocation using Swapping policy.

According to the to the swapping policy lease 2 will be allocated first because it has highest resource requirement of node which is 3.So it will execute in the time slot 12.00 to 12.40.Forwhich lease 1 is unable to meet its deadline. Then lease 3 will be executed within its deadline where as lease 4 cannot be allocated. To overcome this problem Backfilling policy is used.

Allocation using Backfilling policy

Backfilling policy uses rescheduling of leases with preemption. According to the availability of resources multiple leases can be allocated simultaneously. Now the previous scenario can be discussed using Backfilling policy to provide better resource allocation which is shown in the figure-6.

Time → Node ↓	12.00-12.20	12.20-12.40	12.40-1.00	1.00-1.20	1.20-1.30	1.30-1.50
d	3					4
c	3	2	2			4
b	1	2	2	3	3	4
a	1	2	2	3	3	4

Figure 6: Allocation using Backfilling policy.

In case of Backfilling lease 1 will be allocated because it needs 2 nodes and having less execution time. Then simultaneously lease 3 can be allocated where it needs 2 nodes where the 2 nodes are free. Lease 3 will be executed only for time slot 12.00 to 12.20 which is 20 minutes duration out of its 50 minutes duration. Then lease 3 will be allocated completely to meet its deadline. Again lease 3 will be backfilled and allocated in time slot 1.00 to 1.30 for the duration of 30 minutes to meet its deadline. Then lease 4 will be allocated as per the requirement of the nodes and finish the execution with in the deadline.

CONCLUSION AND FUTURE WORK

In this paper we have studied the deadline sensitive based resource allocation policy which is discussed briefly with the suitable example. The problem of the deadline sensitive policy is also studied and overcome by using swapping and backfilling policy. Backfilling policy is also discussed to avoid the problem which is occurred in swapping allocation policy. From the study we observed that swapping is quite simple to implement but all types problem cannot be solved. As compared with swapping policy backfilling policy can solved more problems which cannot be solved by swapping policy. Also backfilling provides better resource utilization as compared to other policies. More examples can be implemented and tested in different scheduling environment in cloud computing for better resource allocation in future work.

REFERENCES

[1] Amit Nathani , Sanjay Chaudhary, Gaurav Somani : "Policy based resource allocation in IaaS cloud" , *Future Generation Computer Systems* 28 ,Elsevier,2012 ,pp 94–103

[2] Mohamed Abu Sharkh, Abdelkader Ouda, and Abdallah Shami,"*A Resource Scheduling Model for Cloud Computing Data centers,"IEEE Communication Magazine,*2013.

[3] Weiwei Lin, Baoyun Peng, Chen Liang ,etc"*Novel Resource Allocation Model and Algorithms for Cloud Computing*" IEEE 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies,2013.

[4] Mansoor Alicherry, T.V. Lakshman" Net work Aware Resource Allocation in Distributed Clouds" 2012 Proceedings *IEEE INFOCOM*, 2012.

[5] Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Capacity leasing in cloud systems using the opennebula engine. In: *Cloud Computing and Applications*, CCA 2008 (2009).

[6] Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: *Resource leasing and the art of suspending virtual machines*. In: 11th IEEE International Conference on High Performance Computing and Communications, pp. 59–68. IEEE (2009).

[7] Sotomayor, B., Keahey, K., Foster, I.: Combining batch execution and leasing using virtual machines. In: Proceedings of the 17th International Symposium on High Performance Distributed Computing, *HPDC* 2008, pp. 87–96. ACM (2008)

[8] B. Khargharia, S. Hariri, M.S. Yousif, *Autonomic power and performance management for computing systems, Cluster Computing* 11 (2) (2008) 167–181.

[9] V. Petrucci, O. Loques, D. Mossé, A *dynamic optimization model for power and performance management of virtualized clusters*, e-Energy'10, ACM, New York, NY, USA, 2010, pp. 225–233. http://doi.acm.org/10.1145/1791314.

[10] M. Bichler, T. Setzer, B. Speitkamp, *Capacity planning for virtualized servers,Presented at Workshop on Information Technologies and Systems*, WITS,Milwaukee, Wisconsin, USA, 2006.

[11] G. Khanna, K. Beaty, G. Kar, A. Kochut, *Application performance management in virtualized server environments, in: Network Operations and Management Symposium*, 2006. NOMS 2006, 10th IEEE/IFIP, pp. 373–381, 2006. http://dx. doi.org/10.1109/NOMS.2006.1687567.

[12] Sotomayor, B., Keahey, K.: Haizea (May 05, 2011),http://haizea.cs.uchicago.edu/

[13] Dhairya Vora1, Sanjay Chaudhary1, Minal Bhise1,et al," *Allocation of Slotted Deadline Sensitive Leases in Infrastructure Cloud*" ICDCIT 2012, LNCS 7154, Springer-Verlag Berlin Heidelberg 2012, pp. 242–252