An Efficient Resource Allocation Algorithm for IaaS Cloud

Sanjaya K. Panda[1] and Prasanta K. Jana[2]

[1] Department of Information Technology
Veer Surendra Sai University of Technology, Burla, India
[2] Department of Computer Science and Engineering
Indian School of Mines, Dhanbad, India
sanjayauce@gmail.com, prasantajana@yahoo.co.in

Abstract. Infrastructure as a Service (IaaS) cloud provides access to computing resources by forming a virtualized environment. The resources are offered by means of leases. However, it is not possible to satisfy all the leases due to finite capacity of resources (or nodes). Mapping between all the leases and the available nodes referred as resource allocation problem is very challenging to IaaS cloud. In this paper, we propose a resource allocation algorithm for IaaS cloud which is based on a novel approach of alert time. First, it uses alert time to assign the leases and then employs swapping to reschedule the already accommodated leases in case a lease is not schedulable by the alert time. This makes resource allocation superior to support the deadline sensitive leases by minimizing the lease rejection in contrast to two existing algorithms by Haizea [3] and Nathani [2]. We perform extensive experiments on several synthetic data sets and the results show that the proposed algorithm outperforms both the algorithms in terms of accepted leases and rejected leases.

Keywords: Resource Allocation, IaaS Cloud, Alert Time, Swapping, Haizea.

1 Introduction

Resource allocation in IaaS cloud is very challenging. The resources are provided to the user on pay-per-use basis [1] and provisioned in the form of virtual machines (VMs) which are deployed on physical machines. The users request such computational resources in the form of leases. The leases are submitted in one of the modes, i.e., AR (advanced reservation), BE (best effort), immediate or DS (deadline sensitive). In AR and immediate mode, resources are non-preemptable and the corresponding leases are time constraint. The BE and DS leases are preemptable and flexible in time constraint. The resource allocation between leases and the VMs (also called nodes) is a well known NP-Complete problem [2]. Therefore, several attempts [2-6] have been made to find a near optimal solution. However, the problem in resource allocation is very crucial and not well studied.

In this paper, we address the same resource allocation problem as described in [2] for IaaS cloud and propose an algorithm called alert time based resource allocation

R. Natarajan et al. (Eds.): ICDCIT 2015, LNCS 8956, pp. 351–355, 2015.

(ALT-RA). The algorithm uses a novel concept based on alert time to assign the resources. The algorithm is tested rigorously with synthetic data sets. The experimental results show that the proposed algorithm performs better than the existing algorithms [2] and [3] in terms of accepted leases and rejected leases.

In the recent years, many resource allocation algorithms [2-6] have been developed for ***Cloud Computing***. Nathani et al. [2] have proposed policy based resource allocation for DS leases in Haizea. But rescheduling and preemption are the major overhead of this system. The algorithm presented in this paper is an improvement over [2] with respect to the following aspects. 1) Our algorithm uses start time and alert time to assign the resource in contrast to submit time and start time as used by [2] and as a result prevention of lease rejection is more. 2) The algorithm uses a novel swapping approach to reschedule the leases (as and when required) instead of sorting the leases in descending order of their resources as used by [2] and this leads reduction of lease swapping.

The remainder of the paper is organized as follows. We describe the resource allocation problem in Section 2. We present the proposed scheme in Section 3 followed by the experimental results in Section 4. Finally, we conclude the paper in Section 5.

2 Problem Statement

We assume here that each lease is a 6-tuple {*ID*, *AT*, *ST*, *D*, *P*, |*N*|} where *ID* denotes the unique identification number, *AT* is the arrival time, *S* is the start time, *D* is the deadline, *P* is the period, |*N*| is the number of nodes. Given a set of *m* resources (called nodes) $N = \{N_1, N_2, N_3, \ldots, N_m\}$ and a set of *n* leases $L = \{L_1, L_2, L_3, \ldots, L_n\}$, the problem of resource allocation is to map the leases onto the available nodes such that the lease rejection is minimized.

Table 1. Submitted leases with their 6-tuple

	ID	*AT*	*ST*	*D*	*P*	*N*	*ALT*
G_1	L_1	08:30	09:00	10:30	40	1	09:50
	L_2	08:35	09:00	10:30	30	3	10:00
	L_3	08:45	09:00	10:40	30	2	10:10
	L_4	08:55	09:00	10:30	40	4	09:50
G_2	L_5	09:25	10:00	11:00	10	1	10:50

We illustrate it with an example as follows. Suppose there are five leases as shown in Table 1. These leases are required to be scheduled with four available nodes. For instance, consider the lease L_3 which requires 2 nodes. The lease can be assigned with one of the following pairs: (N_1, N_2), (N_1, N_3), (N_1, N_4), (N_2, N_3), (N_2, N_4) or (N_3, N_4) where order of nodes in each pair is immaterial.

3 Proposed Scheme

The proposed algorithm is based on the alert time (*ALT*) of the leases which is calculated by using the following Equation.

$$x = D - ST$$
$$ALT = ST + (x - P) - \alpha \quad (1)$$

where α is the overhead time. For example, consider the lease L_1 (refer Table 1) in which the *D* and *ST* values are 10:30 and 09:00 respectively. So, the *x* value is 10:30 – 09:00 = 90. The lease requires a period of 40. Therefore, the *ALT* time is 09:00 + (90 – 40) – 0 = 09:50 assuming that the value of α is zero. The basic idea of the proposed algorithm is as follows. First, it sorts the leases in the ascending order of their *ST*. Then, it divides the whole set of the leases into groups (See Table 1). The leases with the same *ST* are kept in the same group. Next it calculates the alert time of the leases within each group and allocates the leases such that the lease with earliest *ALT* is assigned first. However if there is a tie with the same *ALT* value, then the lease with maximum |*N*| value, i.e., the lease with maximum number of node requirement is assigned first. Note that this approach has two basic advantages over the existing algorithms by Haizea [3] and Nathani et al. [2]: 1) it prevents the deadline of the leases and 2) it utilizes the resources properly. From here onwards we will refer the algorithm of Haizea as HAIZEA and the algorithm of Nathani et al. (without backfilling) as DPS (dynamic planning based scheduling).

3.1 A Typical Case: Resource Allocation with Swapping

This scenario is occurred when the proposed algorithm cannot accommodate a newly arrived lease. Swapping is used to make space for the upcoming leases. So, we propose here a novel swapping approach to reschedule the leases. We first define some terminologies as follows.

Definition 3.1 **(*ST-ALT* lease_set)**: We define it as the set of all already accommodated leases whose duration between ALT and ST intersects with the duration between ALT and ST of a newly arrived lease.

Definition 3.2 **(*ALT-D* lease_set)**: This is the set of all already accommodated leases whose duration between D and ALT intersects with the duration between D and ALT of a newly arrived lease.

The proposed scheme with swapping reschedules the leases by forming two lease_sets, i.e, ST-ALT and ALT-D. Then, it checks whether the intersection of these lease_sets, i.e., {ST-ALT} ∩ {ALT-D} is empty or not. If the intersection is not empty then it finds the difference of these lease_sets using following Equation.

$$\{ST\text{-}ALT\} = \{ST\text{-}ALT\} - \{ALT\text{-}D\} \quad (2)$$

Note that this difference actually makes {*ST-ALT*} ∩ {*ALT-D*} empty. Next, each lease of {*ST-ALT*} is compared with the lease of {*ALT-D*} to decide whether they can be swapped or not.

Remark. If two lease_sets have m leases and l leases, the above comparison takes $O(ml)$ time by the proposed algorithm.

They can only be swapped when the following two conditions are met. First, each {*ST-ALT*} lease has less requested resources than {*ALT-D*}. Second, interchange of {*ST-ALT*} and {*ALT-D*} lease does not violate their deadline constraints. However, if the newly arrived lease is not allocated after swapping, then it rollbacks the rescheduling.

Remark. If the lease_sets has n leases ($n = l + m$), the DPS algorithm takes $O(n^2)$ time in contrast to $O(lm)$ time required by the proposed algorithm.

4 Experimental Results

We tested the proposed algorithm through simulation run with numerous synthetic data sets. The experiments were carried out using MATLAB R2010b on an Intel Core 2 Duo processor, 2.20 GHz CPU and 4 GB RAM running on the platform Microsoft Windows 7. We took four nodes with equal specifications. The numbers of leases were taken as 11, 22, 33 and 44 respectively. However, the parameters of the leases were taken manually. We measured the performance in terms of accepted leases and rejected leases as used by [2]. The accepted and rejected leases are the total number of leases accepted/rejected in a data set. Figs. 1-2 show the comparison of proposed algorithm with the two existing algorithms HAIZEA [3] and DPS [2].

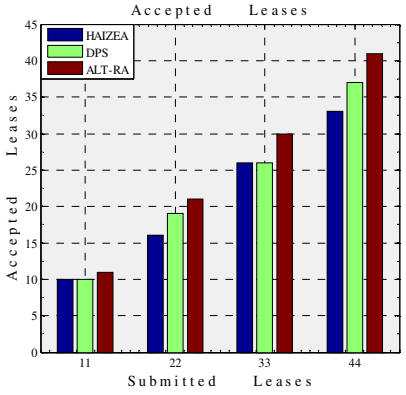

Fig. 1. Accepted leases

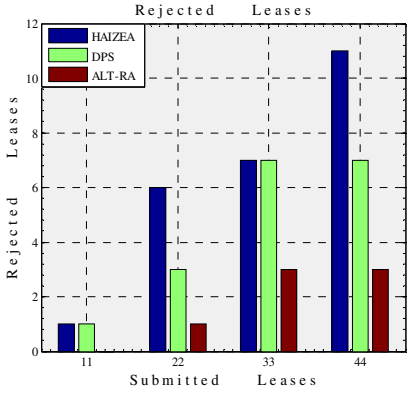

Fig. 2. Rejected leases

5 Conclusion

We have presented a resource allocation algorithm for IaaS clouds. The algorithm is based on alert time of the leases. It was experimented extensively on several synthetic data sets. The experimental results have been compared with two well known existing resource allocation algorithms. The comparison results show that the proposed algorithm outperforms both the algorithms in terms of two performance metrics namely, accepted leases and rejected leases.

References

1. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud Computing and Emerging IT Platforms: Vision, Hype and Reality for Delivering Computing as the 5th Utility. Future Generation Computer Systems 25, 599–616 (2009)
2. Nathani, A., Chaudhary, S., Somani, G.: Policy Based Resource Allocation in IaaS Cloud. Future Generation Computer Systems 28, 94–103 (2012)
3. Haizea, `http://haizea.cs.uchicago.edu/whatis.html` (accessed on January 9, 2014)
4. Vora, D., Chaudhary, S., Bhise, M., Kumar, V., Somani, G.: Allocation of Slotted Deadline Sensitive Leases in Infrastructure Cloud. In: Ramanujam, R., Ramaswamy, S. (eds.) ICDCIT 2012. LNCS, vol. 7154, pp. 242–252. Springer, Heidelberg (2012)
5. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing. Future Generation Computer Systems 28, 755–768 (2012)
6. Akhani, J., Chuadhary, S., Somani, G.: Negotiation for Resource Allocation in IaaS Cloud. In: 4th ACM Conference COMPUTE (2011)