International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-04, Issue-12, Mar 2019

LBSAVM: A Probabilistic Algorithm for Virtual Machine Scheduling

Vivek Shrivastava

International Institute of Professional Studies, Devi Ahilya University, Indore, India.

Abstract **The Infrastructure as a Service (IaaS) is a cloud computing service model. It provides computing resources in the form of virtual machines to consumers. These virtual machines possess computing capacity which is required by cloud client and are provisioned according to the Service Level Agreement (SLA) in the form of lease for a certain period of time. Haizea is very famous open source plug-in scheduler for OpenNebula toolkit. It has Advance Reservation (AR), Best Effort (BE), Immediate, and Dead-line sensitive types of lease. Available literature reveals that BE leases can be queued up for executing in a given order and pre-emptive manner for other higher priority leases. Some BE leases can have a problem of starvation of resources due to lower priority than other BE leases. This paper proposes LBSAVM (Lottery Based Scheduling Algorithm for Virtual Machines) and LSBS (Lottery Scheduling Based Simulator). LBSAVM takes care of starvation problems of BE leases, by providing more chances to lower priority leases. Results based on experimenting with LSBS on proposed policy and implemented algorithm shows significant change in serving lower priority leases and can avoid starvation problem in IaaS cloud environment effectively.**

Keywords – Cloud Computing, IaaS, LABSAVM, Lottery Based Scheduling, Virtual Machine Scheduling.

I. Introduction

The Cloud computing is useful in various domains like Internet of Things, business analytics, machine learning by providing its services in different levels. It provides infrastructure as a service for provisioning computing infrastructure [1]. Software are subscribed and provided over the Internet in cloud computing paradigm as a service, so that there is no need to: purchase software, match them according to available architecture and hardware, install them and then maintain them for a period of time [2]. Operating systems, runtimes, middleware, development platforms are provided over the Internet as a service in platform as a service model of cloud computing [3].

Internet of Things is a next big-wave in computing where each and every device can be connected to Internet and can exchange its information with other computing devices; however these all need to pass their data through IoT gateways [4]. These IoT devices generate ample amount of data but they are very low powered, have very low processing power and storage capacity. These devices are needed to be connected to cloud for storing their data and getting it processed. Sensors as a service can be provided by sensor cloud service providers (SCSP) [5].

Cloud posses an infinite capacity, i.e. it has so many resources that it will never run out of resources. Efficient scheduling is the main key in possessing such property; also elasticity property of cloud requires proper scheduling of resources. Several works have been reported in literature for scheduling resources in such environment.

IaaS cloud environment allows provisioning of VMs to different consumers in the form of leases. Leases have a type called Best Effort (BE) [6]. These BE lease can be pre-emptive in nature, and are pooled in a queue to be served by the scheduler according to some lease attribute like submit time, cost, deadline etc. If a lease is not getting attention of scheduler and not given resources due to other high priority leases, then this is a situation of starvation [7]. Problem of resource starvation in virtual machine scheduling can be solved using efficient scheduling algorithms.

Lottery based scheduling can solve starvation problem successfully [8]. Proposed LBSAVM in this work is a lottery based scheduling algorithm. LBSAVM is a probabilistic scheduling algorithm. It can solve the problem of starvation by assigning every lease to some number of tickets. A lease having more number of tickets has higher chances of selection than other leases with less number of tickets. All leases have at least one ticket, so it guaranties that every lease has non-zero probability of being selected at each scheduling operation.

 DOI : 10.18231/2454-9150.2019.0156

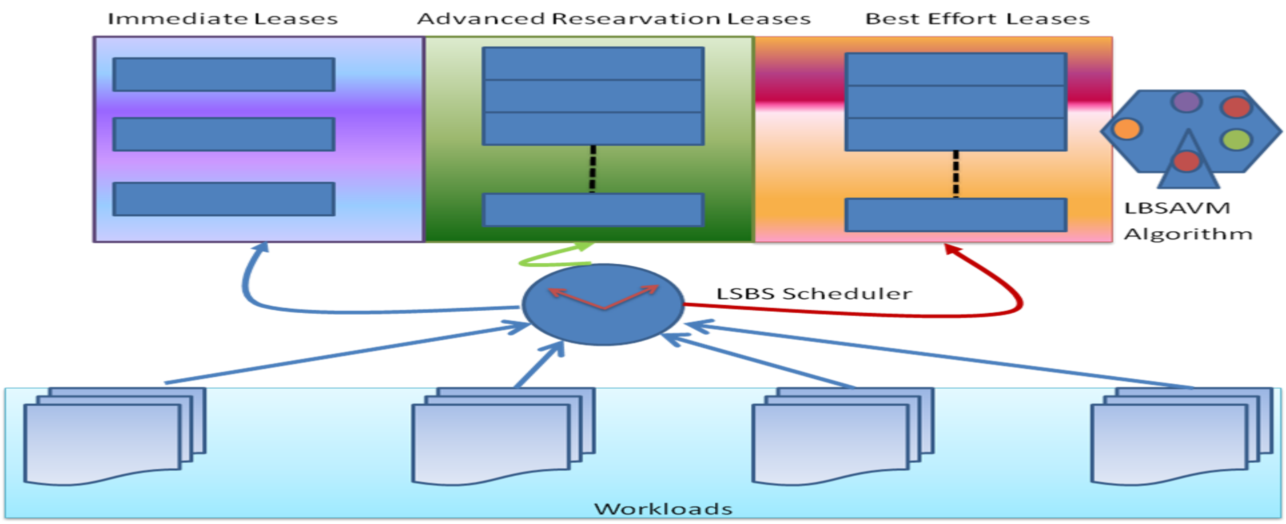

Figure 1 LABSAVM Algorithm, LSBS Scheduler and Workloads

The rest of the paper is organized as follows. Section 2 reviews the available literature. In Section 3, we discuss the Need of LBSAVM Leasing Policy and Algorithm. In Section 4, we have discussed the Proposed Algorithms, which includes the experiments done and results obtained. Section 5 concludes this paper with future enhancements.

II. RELATED WORK

The various researcher groups are investigating the ways in which cloud resources can be scheduled efficiently. Cloud resource scheduling is a nontrivial task because cloud exhibits infinite capacity of resources. Negotiation based reservation of resources has been proposed by the Akhani et *al*. for maximization of resource utilization in [9]. Nathani et *al*. proposed the new type of lease which is deadline sensitive and minimizes rejection rate of newly requesting leases in [10]. Scheduling algorithms like COMMA [11], mEDF [12], CRI [13], Safety [14], [15], IAR [16] and capacity based scheduling [10] have been proposed for cost minimization, profit maximization, migration safely, ranking based, security based and improved advanced reservation leases.

Salehi et *al*. advocated two market-oriented scheduling policies in [17] that intended at satisfying the application deadline by increasing the computational capacity of local resources by the use of hiring resource from Cloud providers.

Buyya et *al*. proposed creation of federated Cloud computing environment (InterCloud) that facilitates just-in-time, opportunistic, and scalable provisioning of application services in [18]. InterCloud proposed by them, can consistently achieve QoS targets under variable workload, resource and network conditions. InterCloud achieved the goal of creating a computing environment that supported dynamic expansion or contraction of capabilities (VMs, services, storage, and database) for handling sudden variations in service demands.

Buyya et *al*. proposed and developed CloudSim toolkit for modeling and simulation of Cloud computing environments in [19]. The CloudSim toolkit supported modeling and creation of one or more virtual machines (VMs) on a simulated node of a Data Center, jobs, and their mapping to suitable VMs. It also allowed simulation of multiple Data Centers to enable a study on federation and associated policies for migration of VMs for reliability and automatic scaling of applications [19].

This paper proposes a new leasing policy and implementation of its algorithm for scheduling the IaaS cloud based resources in a probabilistic manner. This policy and algorithm are capable to schedule resources in an approach that minimizes the starvation of resources to different leases and thus improves the scheduling operations in said environment.

III. NEED OF LBSAVM LEASING POLICY AND ALGORITHM

IaaS model of cloud computing supports multi-tenancy by allowing leases to different consumers. Consumers may have urgency of getting virtual machines or they may need some computing power for some later point of time. Best Effort leases are best suitable for non-interactive workloads i.e. requests are queued up and served when the resources are available in a pre-emptive or co-operative ways. These leases may suffer suspension due to some other high priority leases and in a highly rushed environment some leases can go suspended or in a waiting state for an indefinite period of time due to dynamic nature of cloud.

Proposed Algorithm

Algorithm based on LBSAVM policy is given in this section. System description and supporting modules of LBSAVM are also described here.

A. System Description

A simulator named LSBS (Lottery Scheduling Based Simulator) has been developed for experiments with the

 DOI : 10.18231/2454-9150.2019.0156

proposed LBSAVM policy in this work. This LSBS contains a class named Lease which has data member for lease-identification-number, submit time, requested resources, start, duration, deadline, pre-emptive etc. An array of Lease objects as a random queue is used for assigning lottery tickets and scheduling leases on the basis of winning a lottery. Total number of Lease objects taken is limited in current version to analyze and show the results properly. Only best effort leases with pre-emption are considered in these experiments.

B. Algorithms

In this work, functions for proper scheduling of leases have been developed. These functions include initialization of objects, enqueue lease requests, assign tickets to leases, execute that leases in a simulated clock, store the data about leases in a file etc. initially a random queue is assigned with zeros for showing non-allocation of any lease in that slot, as and when that slot is randomly chosen to fill with random lease id, a zero is replaced with that number. When the lease is executed then that slot is again filled with zero showing vacant slots in random queue.

Following steps have been taken for initialization of lease objects:

1. Execute step number 2 for all leases in leases request pool.

2. Assign a flag variable with value 0 for showing every lease has been given zero ticket.

Following steps have been used in the algorithm for assigning random number of tickets or slots in queue:

1. Execute step number 2 for all leases in leases request pool.

2. Assign random number of tickets to every lease in lease random queue; if it has not been assigned any ticket.

Following steps have been used for selection and execution of leases randomly:

1. Select a lease randomly from the lease random queue.

2. Assign resources to lease selected in step 1 and execute it.

3. Cancel selected ticket from lease selected in step 1.

4. Assign other leases more tickets randomly for increasing their chance of selection next time.

IV. Experiments and Results

A small batch of hundred consumers' VM requests is generated only for checking the validity of our proposed leasing policy and algorithm through LSBS scheduler in lab. Lease submit time, requested resources, start, duration, deadline were generated randomly using rand function available in Java language in a controlled automated environment. Lease id was populated in an auto increment manner for unique identification. LBSAVM policy deals with the starvation problem of resources by using Lottery based scheduling inherently, providing random number of tickets or slots to all lease requests and increasing number of tickets to all leases except the one which has been selected. The graph shown in figure-2 is for a small demonstration, showing randomly generated tickets to lease number 1, 2 and 3 and out of these three leases, selection of one lease for execution. Figure-3 clearly shows the result generated by LSBS scheduler exhibiting decreasing average waiting time and total turnaround time on increasing total number of lottery tickets or slots for a lease in randomqueue.

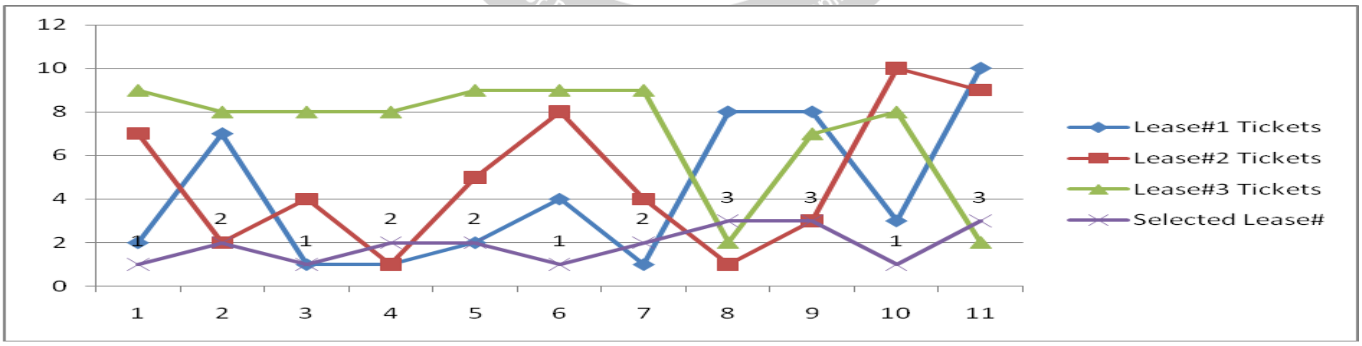

Figure 2: Selection of lease with varying number of tickets allotted to different leases.

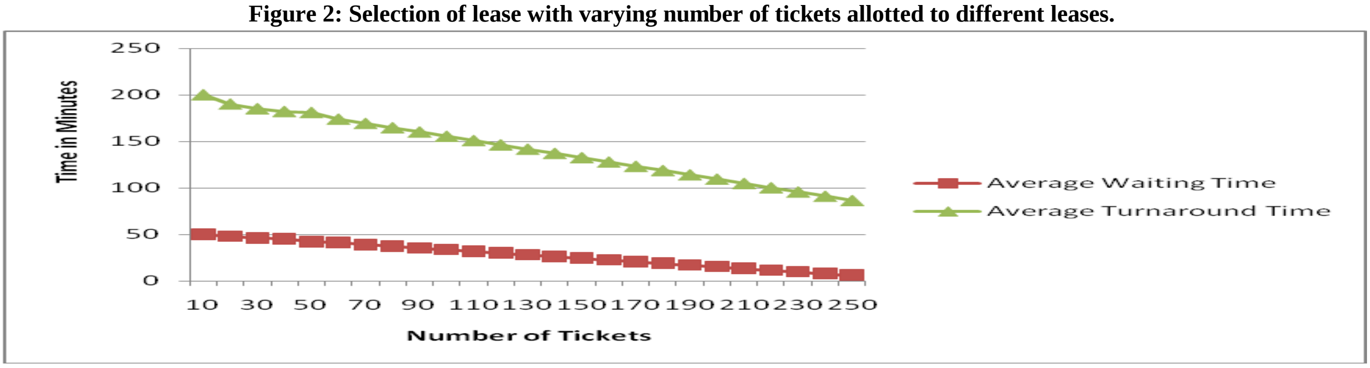

Figure 3: Average Waiting Time and Average Waiting Time of Best Effort Leases.

 DOI : 10.18231/2454-9150.2019.0156

V. CONCLUSION

Best effort leases are very useful in case of non-interactive workloads, but due to lower priority they may suffer starvation problem, so proposed algorithm may be helpful for increasing their priority and get the resources in a considerable duration of time. This algorithm continuously increases the priority of lower priority leases by assigning them more tickets on every suspension of them, or on selection of some other lease in a probabilistic manner. Proposed work opens the challenges of increasing the priority in a deterministic way. Proposed algorithm may be extended for further research in IaaS cloud environment for improved scheduling decisions.

REFERENCES

[1] T. Bhardwaj, , M. Kumar, & S. C.Sharma, Megh: a private cloud provisioning various IaaS and SaaS, Proc. Soft Computing: Theories and Applications, Springer, Singapore.

[2] S. C. Nayak, S. Parida, C. Tripathy, & P. K. Pattnaik, Modeling of Task Scheduling Algorithm Using Petri-Net in Cloud Computing, Khalid Saeed, Nabendu Chaki, Bibudhendu Pati, Sambit

Bakshi, and Durga Prasad Mohapatra, (Ed.), Progress in Advanced Computing and Intelligent Engineering, (Singapore: Springer, 2018) 633-643.

[3] M. A. Rodriguez, and R. Buyya, Scheduling dynamic workloads in multi-tenant scientific workflow as service platforms, Future Generation Computer Systems, 79, 2018, 739-750.

[4] S. Basu, M. Karuppiah, K. Selvakumar, K. C. Li, , S. H. Islam, M. M. Hassan, and M. Z. Bhuiyan. An intelligent/cognitive model of task scheduling for IoT applications in cloud computing environment, Future Generation Computer Systems, 2018.

[5] D. Zhu, IOT and big data based cooperative logistical delivery scheduling method and cloud robot system. Future Generation Computer Systems, 2018.

[6] S. C. Nayak, S. Parida, C. Tripathy, & P. K. Pattnaik, Task Scheduling Mechanism Using Multi-criteria Decision-making Technique, MACBETH in Cloud Computing, in Prasant Kumar Pattnaik, Siddharth Swarup Rautaray, Himansu Das, Janmenjoy Nayak (Ed.), Progress in Computing, Analytics and Networking. Springer, (Singapore: Springer, 2017) 381-392.

[7] V. Shrivastava, and D. S. Bhilare, Algorithms to improve resource utilization and request acceptance rate in IaaS cloud scheduling, International Journal of Advanced Networking and Applications, 3(5), 2012, 1367-1374.

[8] Shyam, G. K., & Chandrakar, I. (2018). Resource Allocation in Cloud Computing Using Optimization Techniques. In Cloud Computing for Optimization: Foundations, Applications, and Challenges (pp. 27-50). Springer, Cham.

[9] J. Akhani, S. Chuadhary, & G. Somani, Negotiation for resource allocation in IaaS cloud, Proc. Fourth Annual ACM Bangalore Conference, Banglore, India, 2011 , 15-21.

[10] A. Nathani, S. Chaudhary, and G. Somani, Policy based resource allocation in IaaS cloud, Future Generation Computer Systems, 28(1), 2012, 94-103.

[11] V. Shrivastava, & D. S. Bhilare, COMMA: A Cost Oriented Market and Migration Aware Leasing Policy and Algorithm in IaaS Clouds, Proc. ACM International Conference on Information and Communication Technology for Competitive Strategies, Udaipur, India, 2014, 52-58.

[12] V. Shrivastava, and D. S. Bhilare, mEDF: Deadline Driven Algorithm for Minimizing Response Time and Completion Time in IaaS Clouds. International Journal of Application or Innovation in Engineering & Management, 3(6), 2014, 16-22.

[13] V. Shrivastava, and D. S. Bhilare, CRI: A Novel Rating Based Leasing Policy and Algorithm for Efficient Resource Management in IaaS Clouds, International Journal of Computer Science and Information Technologies, 5(3), 2014, 4226-4230.

[14] V. Shrivastava, and D. S. Bhilare, SAFETY: A framework for secure IaaS clouds, International journal of Advanced Networking and Applications, 6(6), 2015, 2549-2555.

[15] V. Shrivastava, and D. S. Bhilare, A Security Aware Leasing Policy And Algorithm For IaaS Clouds, International journal of Engineering Research & Technology, 4(6), 2015, 886-891.

[16] V. Shrivastava, P. Gupta, and D. S. Bhilare. IAR: Improved Advance Reservation in IaaS Clouds, International Journal of Computer Applications, 152(2), 2016, 4-9.

[17] M. A. Salehi, & R. Buyya, Adapting market-oriented scheduling policies for cloud computing, Proc. Springer International Conference on Algorithms and Architectures for Parallel Processing, Berlin, Heidelberg, 2010, 351-362.

[18] R. Buyya, R. Ranjan, & RN Calheiros, Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services, Proc. International Conference on Algorithms and Architectures for Parallel Processing, Springer, Berlin, Heidelberg, 2010, 13-31.

[19] R. Buyya, R. Ranjan, & RN Calheiros, Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities, Proc. International Conference on High Performance Computing & Simulation, IEEE, Leipzig, Germany, 2009, 1-11.

 DOI : 10.18231/2454-9150.2019.0156