A Secure VM Consolidation in Cloud Using Learning Automata

Sambit Kumar Mishra, Bibhudatta Sahoo, and Sanjay Kumar Jena

National Institute of Technology, Rourkela, India
{skmishra.nitrkl,bibhudatta.sahoo}@gmail.com,skjena@nitrkl.ac.in

Abstract. Cloud computing system is a progression of distributed system that has been adopted by worldwide scientifically and commercially. For optimal utilization of clouds potential power, effective and efficient algorithms are expected, which will select best resources from available cloud resources for different applications. This allocation of user requests to the cloud resources can optimize several parameters like energy consumption, makespan, throughput, etc. In this paper, we have proposed a learning automata based algorithm to minimize the makespan of the cloud system and also to increase the resource utilization that holds secured resource allocation. We have simulated our algorithm, *AOALA* with the help of CloudSim simulator in a heterogeneous environment. During the comparison of the algorithm, we provide a finite set of tasks to the *AOALA* algorithm once and estimate the makespan of the system. We have compared our proposed technique (*AOALA*), i.e., with learning automata and without learning automata (random allocation algorithm), and shows the system performance.

Keywords: Cloud Computing; DVFS; Learning Automata; Makespan; Resource; Task Allocation; VM;

1 Introduction

Cloud computing is an evolution of Grid Computing. In the current scenario, Cloud computing is a buzzword and acquiring more and more concentration from users. Cloud computing affords a vast pool of shareable resources (CPU, memory, storage, workstations, operating system, energy, network throughput, network loads and delays, etc.) which delivers scalable on-demand resources as a service over the Internet with the cooperation of virtualization technique. In other words, a Cloud is a collection of parallel and distributed system which is interconnected and virtualized. These virtualized resources allotted to the customer according to their respective Service Level Agreement (SLA) between the customer and supplier. The virtualization technique virtualizes the physical resources of the physical hosts in the form of virtual machines (VMs).

In this modern time, every user wants to get their services in less time. Therefore, the task allocation performs a significant role [1]. The number of users managing over the Internet is much more and increasing with day-by-day. To afford services to these huge number of users is a challenging task and one of the best solutions is Cloud Computing [2]. Task consolidation problem is a combinatorial optimization problem in the fields computer science. Allocation of the task in cloud computing is an NP-Hard problem [1]. Different researchers have been worked on this allocation problem and presented various heuristic algorithms [3, 4, 9, 15, 17]. The aim of task consolidation problem is to assign a finite set of tasks to cloud resources in such a way that it will approach to minimize makespan, minimize the execution cost and maximize the utilization of resources [9, 11]. In the current time, cloud computing is earning lots of interest in various domains by processing big data [5, 6]. In that, data are gathered from several sources such as sensor networks, social networks, and vehicles [8]. There is still scope to address security matter of data collected from above sources to cloud data center.

To have a suitable model for a realistic environment, agents need to be able to illustrate the flexible behaviors in various situations, because, in agreements, agents struggle with complex environments, unusual deadlines and inadequate information regarding the opponents [7]. Therefore, an agreement system must be designed with efficient learning tools to be ready to adopt a proper strategy. Learning through setting out the selection of actions of the cloud providers based on the gathered information over time, could satisfy such design. The complication of the learning effect in a multi-agent system more because the agents have to learn the influences of their activities and also, coordinate with other agents. Learning automaton is an adaptive agent applied for delivering decisions [10]. Here, we propose an adaptive resource allocation mechanism where each cloud resource (or VM) has an assigned priority for each task.

The main aim of writing this paper is to introduce a heuristic algorithm for minimizing the makespan and maximizing the resource utilization. We are viewing resources as heterogeneous in nature. The remaining of the paper is constructed as follows. Section 2 outlines a summary of learning automata; Section 3 describes a brief idea about the problem statement. Section 4 describes our proposed algorithm to minimize the makespan of the heterogeneous computing environment along with the secured migration. In Section 5, explains about simulation and results, utilization of VMs and effectiveness of our algorithm followed by the conclusion of the paper in Section 6.

2 Learning Automata

The idea of learning automaton is identified as an importance of the effort on illustrating experimental execution on the arrangement of activities from the past studies. The word learning is the divergence in the system performance dependents upon the past studies with respect to time. The environment generates different solutions to make the automated system. The learning automaton selects one action randomly from a set of actions. The selection will be done randomly because all actions have same probability value. The action probabilities are modified according to the response supplied by the environment. This process is repeated continuously to settled the optimal action by choosing the action with the largest probability value. Learning automata has attained applications in designing automated system, modeling biological system, computer vision, transportation, particle swarm optimization, etc.. Several learning automata

Fig. 1: Interaction of Learning Automata and Environment

and their utilization have studied in [13, 14]. The most significant feature of a learning system is its adaptable capability to magnify the efficiency over time. The system is represented as $E = \{\alpha, \beta, \gamma\}$ in which α is set of inputs received by the environment, β is set of outputs and γ is the penalty value. Fig. 1, presents the relationship between learning automata and the environment.

3 Problem Statement

The task allocation and VM allocation problems play a significant role in the efficient utilization of cloud resources. There are n number of heterogeneous input tasks to the cloud system and m number of heterogeneous VMs in terms of resource capacity (like processing speed, main memory, secondary storage, bandwidth) in the cloud system. The mapping of tasks to the available VMs is termed as task allocation problem, and the allocation of new VM to the available host for the execution of the new task is termed as VM allocation problem. In the proposed method, according to the input task, some probability value is assigned to each VM. Then, according to the overall performance value and the threshold value of the VM, the VM selection will be made for the particular task. The appropriate assignment of new tasks to some VMs is an assignment problem and a well known NP-complete problem [15]. The sub-optimal solution for the mentioned assignment problem with the aim of minimizing the makespan of the system is the objective of this work.

4 Algorithm

The following presented algorithm is used to achieve an optimal allocation of the set of user requests (tasks) and the cloud resources (VMs). Initially, all the VMs have an equal probability which implies that any of the VMs is fairly likely to be preferred. So, the first VM is picked randomly. Then, the overall performance (*op*) of the VM is estimated and compared with the threshold value. The probability value of that VM is improvised if the *op* is more than the threshold otherwise degraded the probability value. Here, α, and β are reward and penalty constants respectively. The learning system examines the threshold of VM with the overall performance of that VM. For each task, if the *op* value is crossed the threshold for a VM, then that VM is discarded by the learning system.

4.1 Security

At the time of designing a moving target (VM migration) protection scheme, we must assure that the enemies are difficult to overcome the actual internal infrastructures however the approach is not confidential [12]. In this scheme, the tasks are initially allocated to some VMs randomly and then according to the objective; the tasks are moving from one VM to another. This migration of VMs and tasks should be secured; this information should secretly available to the cloud service provider only. As long as the enemies cannot reach the information of all cloud resources (VMs), our approach is always safe.

Algorithm 1 : *AOALA*: Algorithm for Optimal Allocation using Learning Automate

Input: T: set of n tasks sorted in descending order of their length; T_j: Threshold set for each tasks; P_{ij}: Action probability of i^{th} task on j^{th} VM;
Output: Mapping of Tasks and VMs

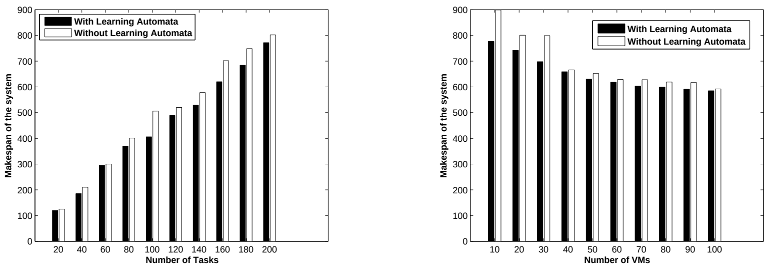
```
1: for each task T_i, 1 ≤ i ≤ n do
2:     Initialize same action probability for all VMs;
3:     Randomly, select one of the VM;
4:     Calculate op_j;
5:     if op_j < T_j then
6:         P_ij(k+1) = P_ij(k) + α(1 − P_ij(k));
7:         P_il(k+1) = (1 − α) × P_il(k), i ≠ j, 1 ≤ l ≤ m;
8:     else
9:         P_ij(k+1) = (1 − β) × P_ij(k);
10:        P_il(k+1) = β/(n−1) (1 − β) × P_il(k), i ≠ j, 1 ≤ l ≤ m;
11:    end if
12:    SV = Select the VM with highest P_ij value;
13:    if |SV| = 1 then
14:        Allocate T_i to V_j;
15:    else
16:        Randomly select a VM from SV;
17:        Goto Step-4;
18:    end if
19: end for
```

We remark that during VM migration, there might be some information leakage. But, in this paper, the design of VM migration systems is not within the range, we prospect more secured VM migration schemes. Virtualization technology provides security services with high assurance by confining them into separate protected VMs. It supports security applications to have perfect visibility to raw memory state of other VMs.

5 Experimental Result

The simulation is conducted on the cloud computing environment tools: CloudSim [16]. CloudSim is one of the simulation tools of cloud environment which provides evaluation and testing of cloud services and infrastructure before the development of the real world. One data center is created with default properties as it mentioned by the CloudSim designer. All hosts are running on a single data center. To simulate our approach, we have some assumptions as follows:

– Total number of VMs will be proportion to the available cores in a host.
– All hosts, as well as VMs, have heterogeneous resource capacity.
– Each host has a finite number of VMs.
– Tasks are non-preemptive in nature.
– Each task is independent in nature, and also the resource requirement of each task is independent of each other.



(a) (b)

Fig. 2: Comparison of makespan of the system using leaning automata $AOALA$ and without using leaning automata, where the number of VM is fixed and task varies (as shown in (a)), and the number of task is fixed and VM varies (as shown in (b)).

In scenario-1, the number of virtual machines is set to 40, and the number of tasks increases from 20 to 200 in the gap of 20. The comparison graph for the calculation of makespan of the system using the learning automata technique ($AOALA$) and without using the learning automata technique is shown in Fig. 2(a). From the fig, the consumption of makespan value in the $AOALA$ algorithm is less. In scenario-2, the number of tasks is set to 100, and the number of virtual machines increases from 10 to 100 in the gap of 10. The comparison graph for the calculation of makespan of the system using the learning automata technique ($AOALA$) and without using the learning automata technique is shown in Fig. 2(b). From the fig, the consumption of makespan value in the $AOALA$ algorithm is less.

6 Conclusion

In this paper, we have studied the learning-based task scheduling approaches in homogeneous and heterogeneous cloud environments proposed by different researchers. We have proposed a task based heuristic algorithm with the help of learning automata to make the system automated for dynamic task allocation in heterogeneous cloud computing environment. We have discussed the learning based system model. We have proposed the $AOALA$ allocation algorithm to implement the mapping of cloud tasks to VMs. The proposed algorithm is simulated in CloudSim simulator, and then we compared the algorithm with learning automata and without learning automata assigning priorities to the VMs. The simulation result shows that $AOALA$ algorithm has improved makespan.

Acknowledgements: This research work is partially supported by Information Security Education and Awareness Project, Phase-II (ISEA-II) funded by Ministry of Electronics and Information Technology (MeitY), Government of India.

References

1. Krishna, P. V.: Honey bee behavior inspired load balancing of tasks in cloud computing environments. In: Applied Soft Computing, pp. 2292-2303, (2013).
2. Devi, C., V. Uthariaraj,: Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Non preemptive Dependent Tasks. The Scientific World Journal, (2016).
3. Misra, S., Krishna, P. V., Kalaiselvan, K., Saritha, V., Obaidat, M. S.: Learning Automata-Based QoS Framework for Cloud IaaS. In: IEEE Transactions on Network and Service Management, No. 1, Vol. 11, pp. 15-24, (2014).
4. Granmo, O. C., Oommen, B. J.: Solving stochastic nonlinear resource allocation problems using a hierarchy of twofold resource allocation automata. In: IEEE Transactions on Computers, 59(4), pp. 545-560, (2010).
5. Puthal, D., Nepal, S., Ranjan, R., Chen, J.: Threats to Networking Cloud and Edge Datacenters in the Internet of Things. IEEE Cloud Computing, 3(3), pp. 64-71, (2016).
6. Puthal, D., Nepal, S., Ranjan, R., Chen, J.: DLSeF: A Dynamic Key-Length-Based Efficient Real-Time Security Verification Model for Big Data Stream. ACM Transactions on Embedded Computing Systems (TECS), 16(2), 51, (2016).
7. Matos, N., Sierra, C., Jennings, N. R.: Determining Successful Negotiation Strategies: An Evolutionary Approach. In: International conference in Paris, france, pp.182-189, (1998).
8. Puthal, D., Mir, Z. H., Filali, F., Menouar, H.: Cross-layer architecture for congestion control in Vehicular Ad-hoc Networks. In IEEE International Conference on Connected Vehicles and Expo (ICCVE), pp. 887-892, (December 2013).
9. Mishra, S. K., Deswal, R., Sahoo, S., Sahoo, B.: Improving energy consumption in cloud. In Annual IEEE India Conference (INDICON), pp. 1-6, (December, 2015).
10. Thathachar, M. A., Sastry, P. S.: Networks of Learning Automata: Techniques for Online Stochastic Optimization. In: New York, NY, USA: Springer, (2004).
11. Sahoo, S., Nawaz, S., Mishra, S. K., Sahoo, B.: Execution of real time task on cloud environment. In Annual IEEE India Conference (INDICON), pp. 1-5, (2015).
12. Zhang, Y., Li, M., Bai, K., Yu, M., Zang, W.: Incentive compatible moving target defense against vm-colocation attacks in clouds. In: IFIP International Information Security Conference, Springer Berlin Heidelberg, pp. 388-399, (2012).
13. Torkestani, J. A., Meybodi, M. R.: Finding minimum weight connected dominating set in stochastic graph based on learning automata. Inf. Sci., vol 200, pp. 5777, (2012).
14. Esnaashari, M., Meybodi, M. R.: Deployment of a mobile wireless sensor network with k-coverage constraint: A cellular learning automata approach. In: Wireless Network, vol. 19, no. 5, pp. 945968, (2013).
15. Li, K., Xu, G., Zhao, G., Dong, Y. and Wang, D.: Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization. In: Sixth Annual China grid Conference, Liaoning, pp. 39, (2011).
16. Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. In: Software: Practice and experience, 41(1), pp. 23-50, (2011).
17. Puthal, D., Sahoo, B. P. S., Mishra, S., Swain, S.: Cloud computing features, issues, and challenges: a big picture. In IEEE International Conference on Computational Intelligence and Networks (CINE), pp. 116-123, (2015).