REVIEW ON VIRTUAL MACHINE RESOURCE ALLOCATION IN CLOUD-FOG COMPUTING SYSTEMS

NIKITHA S.M[1], Dr ANITHA V[2], Dr K L SUDHA[3]

[1]M.Tech Student, ECE, DSCE, Bengaluru
[2]Professor, ECE, DSCE, Bengaluru
[3]PG Coordinator, ECE, DSCE, Bengaluru

***

ABSTRACT:- Cloud is a resource pool from where resources can be accessed. Fog computing has been proposed to deal with a vast number of connected devices. As the resources are finite, it is very difficult to fulfill all demanded resources by the consumer and so that it is not possible to meet cloud consumers QOS requirement. Virtual machine allocation is allocation of a set of virtual machines to a set of physical machines located on data centers. Here the objective is to increase the revenue by increasing the resource usage by allocating the VM in the efficient manner so that resource utilization will get increase and hence more revenue will be generated. This paper provides a view on the current state research to allocate virtual machine in the area of fog computing and internet of things technology.

Keywords: cloud computing, fog computing, virtual machine allocation, internet of things

1. INTRODUCTION

Cloud computing can be an efficient alternative to owning and maintaining computer resources and applications for many organizations, particularly small- and medium- sized organizations, due to the pay-as-you-go model and other characteristics e.g., on-demand, self-service, resource pooling and rapid elasticity. The continued interest in cloud computing has also resulted in other emerging cloud paradigms, such as fog computing.

In fog computing, cloud elastic resources are extended to the edge of the network, such as portable devices, smart objects, wireless sensors and other Internet of Things (IOT) devices to decrease latency and network congestion. IOT devices use interconnected technologies like Radio Frequency Identify (RFID) and Wireless Sensor and Actor Networks (WSAN) to exchange information over the Internet, and are more integrated in our daily life. Smart-home, smart-city and smart-grid are examples of IOT applications, where sets of sensors are used to obtain information to improve the quality of life and quality of experiences. IOT is characterized by widely distributed objects known as ‘‘things’’ with limited storage and processing capacity to guarantee efficiency, reliability and privacy. However, its applications require geo-distribution, mobility support, location-awareness and low latency to efficiently collect and process data from IOT devices. This information is then used to perform detection and prediction for optimization and timely decision-making process.

Over the past few years, the idea of virtualization technology has become a more common phrase among IT professionals. The main concept behind this technology is to enable the abstraction or decoupling of application payload from the underlying distributed physical host resource. This simply means that the physical resources can be presented in the form of either logical or virtual resources depending on individual choices. Furthermore, some of the advantages of implementing virtualization technology are to assist cloud resource providers to reduce costs through improved machine utilization, reduced administration time and infrastructure costs.

To implement the concept of virtualization cloud developers often adopted and make use of the concept of an open source software framework for cloud computing that implements what is commonly referred to as Infrastructure as a Service (IaaS). This software framework is known as Hypervisor. A hypervisor, also called virtual machine manager (VMM), is one of many hardware virtualization techniques that allow multiple operating systems, termed guests, to run concurrently on a host machine. However, there are different infrastructures available for implementing virtualization for which we have different virtual infrastructure management software for that. In practice, the computing resources in local fogs are usually not as abundant as those in remote clouds. When a large number of user terminals offload their applications to a local fog, the fog may use up its resources such that new requests have no chance to be admitted. Therefore, the coordinated VM allocation for the remote cloud and the local fog is an effective approach to meet the requirements of users.

2. RELATED WORK.

Most of the existing VM allocation methods focus on planning algorithms which mainly include static optimization algorithms and dynamic optimization algorithms. In [1], heterogeneous resource (central data centers and pervasive mobile devices) sharing problem if formulated and solved it via convex optimization approaches. An optimal workload allocation problem for the fog and the cloud is tackled using an approximate approach by decomposing the primal problem into sub problems [2].

The problem related to computing resource allocation in three-tier IOT fog networks was overcame using a joint optimization approach which combines Stackelberg game and matching[3]. The three-tier IOT fog networks include fog nodes, data service operators, and data service subscribers. Rodrigues et al. [4] and [5] presented service delay minimization methods in cloudlet systems through VM migration and transmission power control. Guo and Liu [6] presented energy-efficient computation offloading strategies for multi-access edge computing over fiber-wireless networks.

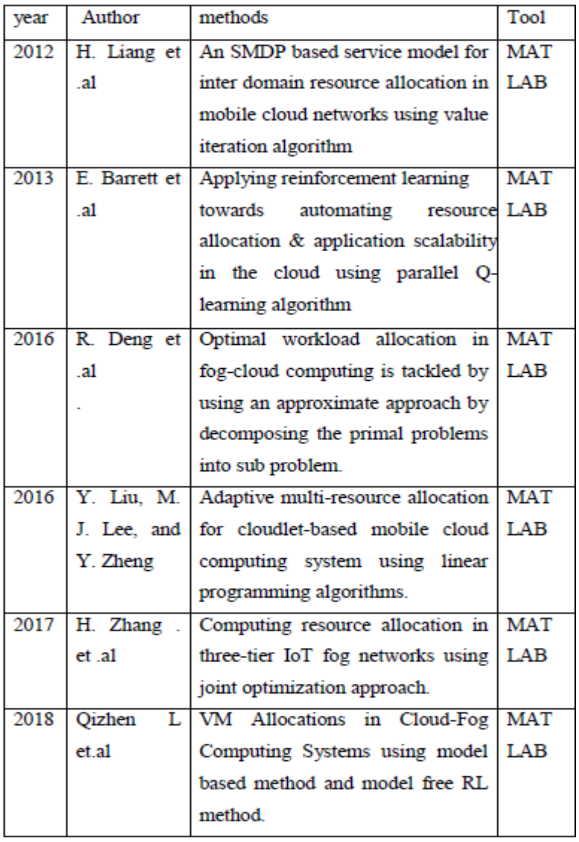
In [7], mobile devices, as decision-makers, predicted wireless bandwidth and cloudlet resources and made offloading decisions. Liang et al. [8] presented a semi-Markov decision process (SMDP) based model for inter domain VM allocation in mobile cloud computing networks and solved the problem using the value iteration algorithm. Li et al. [9] proposed an SMDP-based resource allocation scheme in cognitive enabled vehicular ad hoc networks.

SMDPs were also used in [10] and [11] to establish the dynamic resource allocation models, and linear programming algorithms were used to find an optimal resource allocation strategy under the blocking probability constraint. Hoang et al. [10] took the local cloudlet into account, while Liu et al. [11] considered the joint computing resource allocation for the remote cloud and the cloudlet.

The above planning algorithms are model-based, i.e., system models need to be obtained before planning algorithms are executed. To simplify training the models, some strong assumptions need to be made in the model-based planning methods. The model-free VM allocation problems in which no assumptions on the transition probabilities are made can be solved using reinforcement learning (RL) [12]. However, only a few articles have adopted RL method for VM allocation in cloud computing [13]–[15], and there are even less references using RL for fog computing and mobile cloud computing which are very different from the traditional cloud computing [16], [17]. Rao et al. [15] proposed an RL approach to automate the VM configuration process, where the model is trained with the previously collected samples using supervised learning. In [14], VMs and resident applications were coordinately configured by a model-free hybrid RL approach. Barrett et al. [15] applied the parallel Q-learning algorithm to obtain an optimal computing resource allocation policy. Alam et al. [16] proposed a basic block offloading mechanism based on distributed RL in the mobile fog. Hoang et al. [17] proposed using the policy gradient method which can be viewed as one of the RL algorithms to solve the admission control problem in cloudlets.

In addition, Cao and Cai [18] proposed a game-theoretic machine learning approach to solve the distributed multiuser computation offloading problem for cloudlet-based mobile cloud computing systems Table 1 gives the clear picture of related work done in the field of virtual machine allocation in cloud-fog computing system.

Table 1: comparison of related work



year	Author	methods	Tool
2012	H. Liang et .al	An SMDP based service model for inter domain resource allocation in mobile cloud networks using value iteration algorithm	MAT LAB
2013	E. Barrett et .al	Applying reinforcement learning towards automating resource allocation & application scalability in the cloud using parallel Q-learning algorithm	MAT LAB
2016	R. Deng et .al .	Optimal workload allocation in fog-cloud computing is tackled by using an approximate approach by decomposing the primal problems into sub problem.	MAT LAB
2016	Y. Liu, M. J. Lee, and Y. Zheng	Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system using linear programming algorithms.	MAT LAB
2017	H. Zhang . et .al	Computing resource allocation in three-tier IoT fog networks using joint optimization approach.	MAT LAB
2018	Qizhen L et.al	VM Allocations in Cloud-Fog Computing Systems using model based method and model free RL method.	MAT LAB

3. TERMINOLOGY

Here the virtual machine allocation processes is modeled as a SMDP where the time distribution to next decision approach and state at that time depend on past only through the choice of state and action at the current decision approach. The terminologies used in the current work are as follows:

a) Decision epoch: Here it is a time interval at which the decisions are taken. A time interval between two adjacent decision epochs can be a duration with a random length within [0,∞], so that it can promptly process service requests compared with a discrete-time MDP.
b) State Space: State is defined as the Agent's current Location. The state space is the set of all the available decision-making states.
c) Action space: The action space is the set of all possible actions. When the arrival event of a high priority service request occurs, one of the following actions must be chosen by the service controller:

$$a=\begin{cases}2 & \text{request is handeled by remote cloud}\\ 1 & \text{request is handeled by fog}\\ 0 & \text{rejects request}\end{cases}$$

To select 2 in action space the state has to satisfy the constraint $(n_h^c+1)N_h \leq N_c$, To select 1 in action space the state has to satisfy the constraint $(N_h^f+1)V_h+n_l^f V_l \leq N_c$. If it does not satisfy any of these conditions then action 0 will be selected. When the arrival event of a low priority service request occurs, the service controller must choose one of the following actions:

$$a=\begin{cases}1 & \text{request is handeled by fog}\\ 0 & \text{rejects request}\end{cases}$$

d) Reward Function: The reward function between two consecutive decision epochs can be formulated as

$$R\ (s,\ a,\ j) = k(s,\ a) - \tau\ (s,\ a,\ j)c(s,\ a,\ j)$$

Where k(s, a) is a lump sum reward received by the computing service provider, τ (s, a, j) and c(s, a, j) represent the time interval and the cost rate between two consecutive decision epochs, respectively.

e) Policy (π): The strategy that the agent employs to determine next action based on the current state.

f) Value (V): The expected long-term return with discount, as opposed to the short-term reward R. Vπ(s) is defined as the expected long-term return of the current state sunder policy π.

g) Q-value or action-value (Q): Q-value is similar to Value, except that it takes an extra parameter, the current action a. Qπ(s, a) refers to the long-term return of the current state S, taking action a under policy π.

h) Discount factor(γ)

The discount factor determines the importance of future rewards. A factor of 0 will make the agent "myopic" (or short-sighted) by only considering current rewards, while a factor approaching 1 will make it strive for a long-term high reward. If the discount factor meets or exceeds 1, the action values may diverge. For $\gamma = 1$ without a terminal state, or if the agent never reaches one, all environment histories become infinitely long, and utilities with additive, undiscounted rewards generally become infinite. Even with a discount factor only slightly lower than 1, Q-function learning leads to propagation of errors and instabilities when the value function is approximated with an artificial neural network. In that case, starting with a lower discount factor and increasing it towards its final value accelerates learning.

i) Learning rate (α).

The learning rate or step size α determines to what extent newly acquired information overrides old information. A factor of 0 makes the agent learn nothing (exclusively exploiting prior knowledge), while a factor of 1 makes the agent consider only the most recent information.

4. PROPOSED METHODOLOGY

In the current context 2 methods are going to be used to allocate virtual machine i.e Q and Q (λ) methods. These two are model free methods. These are type of RL algorithm. In both algorithms bellman's optimality equation as in equation 1 is used to update Q- value. In Q (λ) to learn the values of the state value function TD (λ) methods are used.

$$\text{New } Q(s,a)= Q(s,a)+\alpha[R(s,a)+\gamma \max Q'(s',a') - Q(s,a)]\text{----}(1)$$

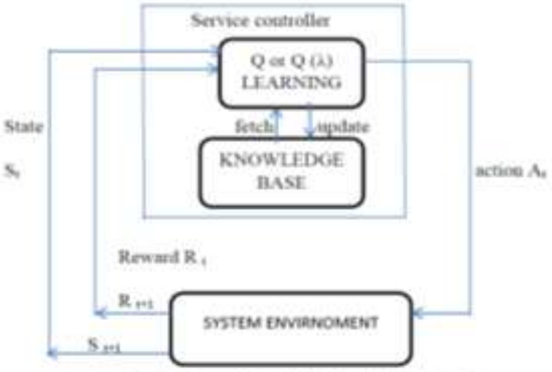


Fig 1: schematic model of Q and Q (λ) algorithm

The proposed methodology adopted in the present project work is depicted in the Figure 1. Figure 1 shows schematic model of Q and Q (λ) algorithm which consists of four elements they are as follows

- Service controller: The service controller contains a learning algorithm processor and a knowledge base. The learning algorithm processor is used to perceive the environmental response, namely extract the state and receive the reward, and update the knowledge base and decision policy according to the immediate environmental response and the data in the knowledge base. The knowledge base stores experience data which can be tabular action values, the connection weights of artificial neural networks, or the weights of linear function, etc. Here assume that knowledge base is made up of the tabular action values.
- System environment: The system environment of the cloud fog computing system consists of the remote cloud, the local fog, and the user terminals within the coverage of the edge node.
- A set of actions: the set of all possible action is called action space. Decision of changing action is taken as discrete time intervals.
- The environmental response.

The Q and Q (λ) algorithm is given below.

A.

-algorithm:

```
1.  Initialize Q, α,λt
2.  Observe state
3.  If request arrives, check whether λ<λt or λ>λt
4.  If λ>λt high priority request. Choose action in action
    space {2,1,0} and perform action with respect to choice.
5.  Update Q
    Else
    choose action in action space{1,0}
    update Q
    end if
6.  Set current state to next and update learning time.
7.  Else
    Update r and t

  End if
```

B. Q(λ) algoritms.

```
1.  Initialize Q(s,a), α,λt,W and e(s,a)for all s and a.
2.  Observe state
3.  Repeat
4.  Extract the features and get feature vector F
5.  If request arrives, check whether λ<λt or λ>λt
6.  If λ>λt high priority request. Choose action in action
    space{2,1,0} and perform action with respect to
    choice. compute reward $R=r+r^l$
7.  Else choose action in action space {1,0} and perform
    action with respect to choice. compute reward
    $R=r+r^l$
8.  End if
9.  Choose $a^l$ from $s^l$ using ε-greedy
10. Then $a^*$= arg max $Q(s^l,b)$,
11. $\delta = R + \gamma Q(s^l, a^*) - Q(s,a)$
12. e(s,a) = e(s,a) + 1
13. For all s , a
14. Update Q(s,a)
15. If ($a^l = a^*$)
16. e(s,a) = $\gamma\lambda$e(s, a)
17. Else
18. e(s,a) = 0
19. s = s′, a = a′.
```

5. ADVANTAGES OF Q and Q(λ) ALGORITHM:

As Q and Q (λ) methods can be employed to allocate VM so the advantages of these methods are follows:

- These methods can be more flexible for learning off-policy.
- These methods considered the model as continuous.
- Here the services are provided based on the priority bases. High priority requests are will be not rejected.

6. CONCLUSION:

Q

Cloud-fog Computing is the new era of computing for delivering computing as a resource. The success and beauty behind cloud computing is due to the cloud services provided with the cloud. Due to the availability of finite resources, it is very important for cloud providers to manage and assign all the resources in time to cloud consumers as their requirements are changing dynamically. So in this paper the problem of virtual machine allocation with its different techniques in cloud computing environments has been considered.

7. REFERENCE

[1] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in Proc. ACM 1st Int. Workshop Mobile Cloud Comput. Netw., Bengaluru, India, Jul. 2013, pp. 19–26.

[2] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," IEEE Internet Things J., vol. 3, no. 6, pp. 1171–1181, Dec. 2016.

[3] H. Zhang et al., "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching," IEEE Internet Things J., vol. 4, no. 5, pp. 1204–1215, Oct. 2017.

[4] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," IEEE Trans. Comput.,vol. 66, no. 5, pp. 810–819, May 2017.

[5] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "A PSO model with VM migration and transmission power control for low service delay in the multiple cloudlets ECC scenario," in Proc. IEEE Int. Conf. Commun., Paris, France, May 2017, pp. 1–6.

[6] H. Guo and J. Liu, "Collaborative computation offloading for multi-access edge computing over fiber-wireless networks," IEEE Trans. Veh. Technol., to be published.

[7] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," IEEE Trans. Mobile Comput.,vol. 14, no. 12, pp. 2516–2529, Dec. 2015.

[8] H. Liang, L. X. Cai, D. Huang, X. Shen, and D. Peng, "An SMDP based service model for interdomain resource allocation in mobile cloud networks," IEEE Trans. Veh. Technol., vol. 61, no. 5, pp. 2222–2232, Jun. 2012.

[9] M. Li, L. Zhao, and H. Liang, "An SMDP-based prioritized channel allocation scheme in cognitive enabled vehicular ad

hoc networks," IEEE Trans. Veh. Technol., vol. 66, no. 9, pp. 7925–7933, Sep. 2017.

[10] D. T. Hoang, D. Niyato, and P. Wang, "Optimal admission control policy for mobile cloud computing hotspot with cloudlet," in Proc. IEEE Wireless Commun. Netw. Conf., Shanghai, China, Apr. 2012, pp. 3145–3149.

[11] Y. Liu, M. J. Lee, and Y. Zheng, "Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system," IEEE Trans. Mobile Comput., vol. 15, no. 10, pp. 2398–2410, Oct. 2016.

[12] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction.Cambridge, MA, USA: MIT Press, 1998.

[13] J. Rao, X. Bu, C.-Z. Xu, L. Wang, and G. Yin, "VCONF: A reinforcement learning approach to virtual machines auto-configuration," in Proc. ACM 6th Int. Conf. Auton. Comput., Barcelona, Spain, Jun. 2009, pp. 137–146.

[14] X. Bu, J. Rao, and C.-Z. Xu, "Coordinated self-configuration of virtual machines and appliances using a model-free learning approach," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 4, pp. 681–690, Apr. 2013.

[15] E. Barrett, E. Howley, and J. Duggan, "Applying reinforcement learning towards automating resource allocation and application scalability in the cloud," Concurrency Comput. Pract. Exp., vol. 25, no. 12, pp. 1656–1674, Aug. 2013.

[16] M. G. R. Alam, Y. K. Tun, and C. S. Hong, "Multi-agent and reinforcement learning based code offloading in mobile fog," in Proc. IEEE Int. Conf. Inf. Netw., Kota Kinabalu, Malaysia, Jan. 2016, pp. 285–290.

[17] D. T. Hoang, D. Niyato, and L. B. Le, "Simulation-based optimization for admission control of mobile cloudlets," in Proc. IEEE Int. Conf. Commun., Sydney, NSW, Australia, Jun. 2014, pp. 3764–3769.

[18] H. Cao and J. Cai, "Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach," IEEE Trans. Veh. Technol, vol. 67, no. 1, pp. 752–764, Jan. 2018.

[19] Qizhen Li "SMDP-Based Coordinated Virtual Machine Allocations in Cloud-Fog Computing Systems", IEEE internet of things journal, vol. 5, no. 3, june 2018.

[20] Jianhong Zhang, Ying Shi, and Xiaofei Xie "The Q(λ) Algorithm Based on Heuristic Reward Function", Dalian, China, International Conference on Intelligent Control and Information Processing, August 13-15, 2010