VIRTUAL RESOURCE ALLOCATION IN CLOUD-FOG SYSTEMS USING Q LEARNING METHODS

[1]Nikitha S M,[2]Dr. Anitha V

[1]Student,MTech,Digital electronics and Communication,[2]professor
[1]Electronics and communication,[2] [1]Electronics and communication
[1&2] Dayananda Sagar College of Engineering, Bangalore, Karnataka, India

Abstract: In recent years Internet of Things (IOT) has developed rapidly because of its attractive applications. Due to limited computing capabilities of IOT devices its applications should be offloaded to cloud and fog. This can be achieved by using model free priority based Q learning method, which finds optimal coordinated virtual resource policy in the sense that it maximizes the expected value of the total reward over all successive steps, starting from the current state by considering continuous task. The common objective of the proposed methods is to find an asymptotically optimal resource allocation policy to maximize the revenue from the perspective of the service system. This method overcomes the rejection probabilities of the high priority requests.

Index Terms - ***IOT, cloud, fog, Q learning.***

1. INTRODUCTION.

IOT terminal devices have limited computing capabilities due to the requirements of the deployment costs and the energy consumption. Therefore, a lot of applications in IOT terminal devices must be offloaded to remote clouds to be processed. However, the conventional cloud computing paradigm is insufficient because offloading applications to remote clouds needs multi hop information transfer in wide area networks (WANs), which can cause problems for latency-sensitive applications, such as real-time IOT analytics.

Fog computing, also referred to as edge computing or cloudlet computing, is an efficient way to improve the delay performance of applications and reduce the congestion of WANs. Computing capabilities of fogs are more powerful than those of user terminals. The communication links from user terminals to fogs are much shorter than those from user terminals to remote clouds. Virtualization and application partitioning are two key technologies to improve the quality of service (QoS) of cloud-fog computing systems. Virtualization can facilitate the heterogeneous computing resource management. By abstracting and slicing the physical computing resources (such as central processing units (CPU) time) into virtual machines (VMs), the hardware, feature, and platform heterogeneity can be reduced. Application partitioning can improve the offloading efficiency of service requests. Offloading entire applications to clouds is not always beneficial to mobile users in terms of energy consumption, especially when the offloading process involves intensive communications. Therefore, applications should be partitioned and the subcomponents with high computation load but low communication load should be offloaded to clouds or fogs [2].

Q-learning is a form of model-free reinforcement learning. It can also be viewed as a method of asynchronous dynamic programming (DP)[3]. It provides agents with the capability of learning to act optimally in Markovian domains by experiencing the con-sequences of actions, without requiring them to build maps of the domains.

2. RELATED WORKS.

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 Virtual resource allocation for the remote cloud and the local fog is an effective approach to meet the requirements of users.

But the present Virtual resource allocation methods depends on planning algorithms which mainly include static optimization algorithms [3] [8] and dynamic optimization algorithms [9]–[11].Nishio et al. [3] proposed a heterogeneous resource (central data centers and pervasive mobile devices) sharing problem and solved it via convex optimization approaches. Deng et al. [4] formulated an optimal workload allocation problem for the fog and the cloud, and tackled it using an approximate approach by decomposing the primal problem into sub-problems. Zhang et al. [5] solved the computing resource allocation problem in three-tier IoT fog networks using a joint optimization approach which combines Stackelberg game and matching. The three-tier IoT fog networks embody fog nodes, data service operators, and data service subscribers. Rodrigues et al. [6] and[7] presented service delay minimization methods in cloudlet systems through VM migration and transmission power control.

Guo and Liu [8] presented energy- efficient computation offloading strategies for multi-access edge computing over fiber-wireless networks. In [9], mobile devices, as decision-makers, predicted wireless bandwidth and cloudlet resources and made offloading decisions. Liang et al. [10] presented a semi-Markov decision process (SMDP) based model for interdomain VM allocation in mobile cloud computing networks and solved the problem using the value iteration algorithm. Li et al. [11] proposed an SMDP-based resource allocation scheme in cognitive enabled vehicular ad hoc networks. SMDPs were also used in [12] and [13] 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. [25] took the local cloudlet into account, while Liu et al. [26] considered the joint computing resource allocation for the remote cloud and therefore the cloudlet.

3.PROPOSED METHODOLOGY

3.1 SYSTEM MODEL.

Figure1 shows cloud fog computing system for three priorities.

- High priority service request: once a high priority service request arrives, the controller can provide service by the remote cloud without high delay and transmission cost,
- Medium priority service request: once a medium priority service request arrives, the controller can provide service by the remote cloud without high delay and transmission cost, or transfer the request to the local fog, or reject it.
- Low priority service request: the low priority service requests must either be handled by the local fog or be rejected.

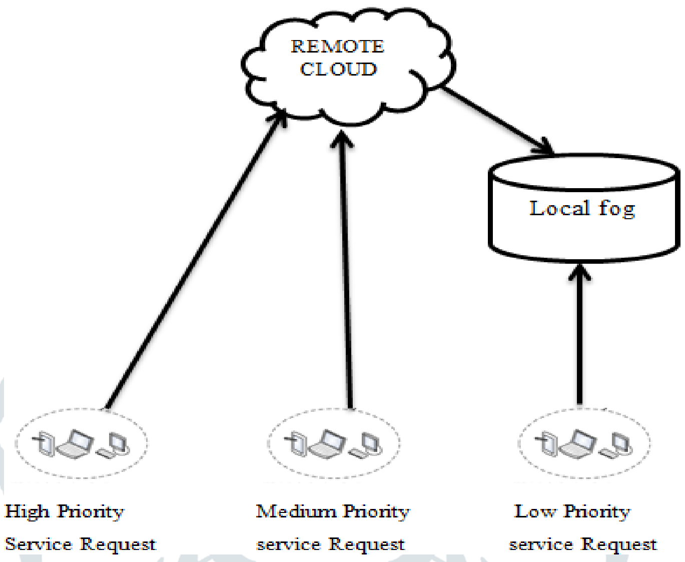

REMOTE CLOUD
Local fog
High Priority Service Request
Medium Priority service Request
Low Priority service Request

Fig 1: system model for cloud-fog computing systems

3.2 SCHEMATIC OF Q LEARNING METHOD

Semi Markov Decision Policy are used in modeling random control problems arising in dynamic systems where the stopover time in each state is a general continuous random variable. Whenever the outcomes are partly random it provides a framework for decision making with the help of decision maker. Semi Markov Decision Process is useful for optimizing problems solved via reinforcement learning. In our Project this policy helps in deciding which outcome is of the highest priority and assign it first to the channel and then the rest based on availability. There is a reward or cost structure associated with the states and decisions, and an information pattern available to the decision maker. In SMDP for every positive outcome the machine gets a reward based on this reward model we can update and assign the services which uses Reinforcement Learning.

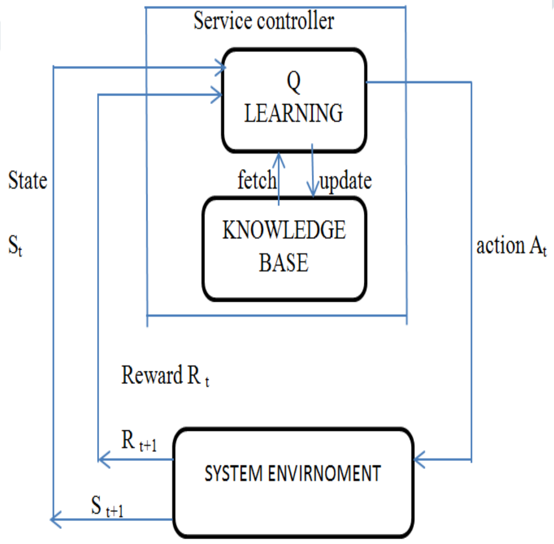


Fig 2: schematic model of Q learning method.

3.3 TERMINOLOGY

Here it is necessary to model the process of virtual resource allocation as a SMDP where the time distribution to next decision approach and state at that time depend 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 $[0,\infty]$, so that it can Promptly process service requests compared with a discrete-time MDP [1].

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 high priority service request occurs, one of the following actions must be chosen by the service controller:

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

When the arrival event of a medium priority service request occurs, one of the following actions must be chosen by the service controller:

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

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: here reward function is considered between two consecutive decision epochs can be formulated as R (s, a, j) = k(s, a) − τ (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. Policy (π): The strategy that agent employs to determine next action based on current state [1]

e) Value (V): It is defined as the long-term return of the current state under policy π which was expected.

f) Q-value: 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.

g) Discount factor (α): The discount factor determines the importance of future rewards. The value of Discount factor can lies between 0 to 1. If the discount factor exceeds 1, the values may diverge so it's better to consider it as 0.6

h) Learning rate (α): It determines to what extent newly acquired information overrides old information. For each iteration if learning is occurring means its value will increase. Consider the learning rate is 0.01.

3.4 PRIORITY BASED Q LEARNING ALGORITHM.

```
1.  Initialization: let m, Q(s , a),a[i],Rand t=0 $\alpha$=0 , $maxQ(s,a) = 1$ and $\gamma$=0.6
2.  While t<$t_{max}$ .monitor the occurrence of events
3.  Whether  request arrives, a[i]>0,check a[0] a[1],increment $\alpha$ by 0.01
4.  if a[0] a[1]==11,it is high priority request ,should be accepted by cloud  and assign R=R+2
5.  elseif a[0] a[1]==10,it is medium priority request ,should be accepted by cloud and assign R=R+2  if it is free otherwise
    local fog should accept and assign R=R+1
6.  else if a[0] a[1]==00,it is low priority request ,should be accepted by local fog and assign R=R+1
7.  else it is invalid request and should be rejected. Assign R=R+0
8.  end if
9.  Update Q(s,a),m to m+1 and t
    $Q(s,a)^1 = Q(s,a) + \alpha[R + \gamma\ maxQ(s,a) - 1]$.
10. Update m and t
```

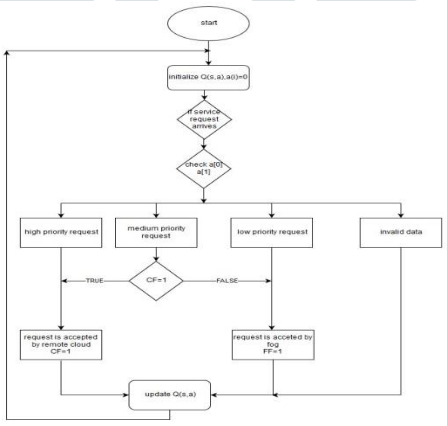
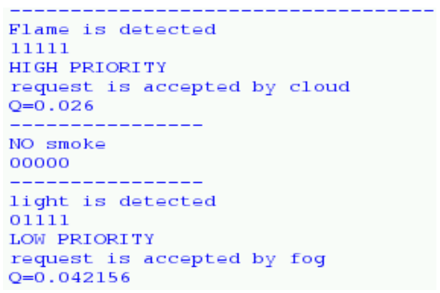


Fig 3.: Flow chart

4.SIMULATION RESULTS.

In this section, in order to evaluate the performance of designed algorithm, python code is written and executed using python IDE. Raspberry pi model B is used as processor so as to receive the inputs from 3 prioritized sensors i.e.,flame sensor as high priority, smoke sensor as medium priority and LDR sensor as low priority request. Here cloud is used for data storage, let's consider thingspeak as cloud. "ThingSpeak is an open-source Internet of Things (IOT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network.

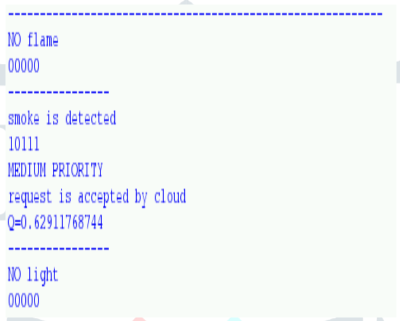
whenever the flame is detected, 11111 is given as input to the processor which will recognize that the incoming input is of high priority and should be accepted by cloud. Whenever Light is detected, 01111 is given as input to the processor which will recognize that the incoming input is of low priority and should be accepted by fog. Figure 4 shows the Simulation of High and low priority sensors.



```
-----------------------------------
Flame is detected
11111
HIGH PRIORITY
request is accepted by cloud
Q=0.026
----------------
NO smoke
00000
----------------
light is detected
01111
LOW PRIORITY
request is accepted by fog
Q=0.042156
```

Fig 4: simulation results of High and Low priority sensors

Whenever smoke is detected, 10111 is given as input to the processor which will recognize that the incoming input is of medium priority and should be accepted by cloud if it is free or else it should be accepted by fog. Figure 5.7 shows the Simulation of medium priority sensors.



```
------------------------------------------------------------
NO flame
00000
---------------
smoke is detected
10111
MEDIUM PRIORITY
request is accepted by cloud
Q=0.62911768744
---------------
NO light
00000
```

Fig 5:simulation results of Medium priority sensors

The sensed data is moved and stored in the cloud. The calculated Q value is also stored along with sensor data. The figure 6,7 and 8 shows the sensor data along with calculated Q value of flame sensor, smoke sensor and LDR sensor respectively. Here the cloud is groped into six fields in which each sensor has two fields, one for sensor data and another for calculated Q value.

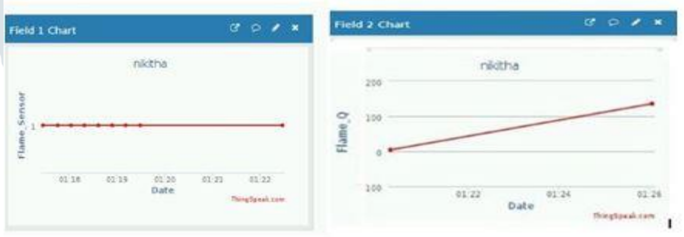


Fig 6: Flame sensor data in cloud

.

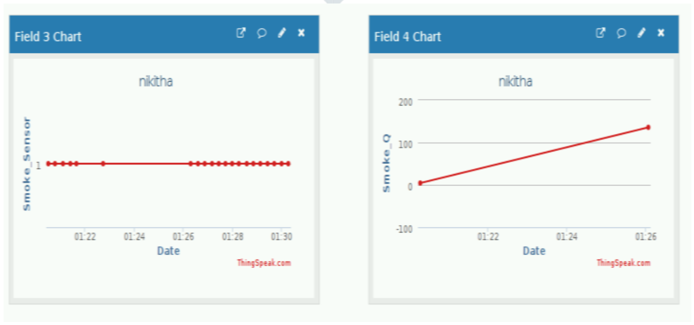


Fig 7: Smoke sensor data in cloud

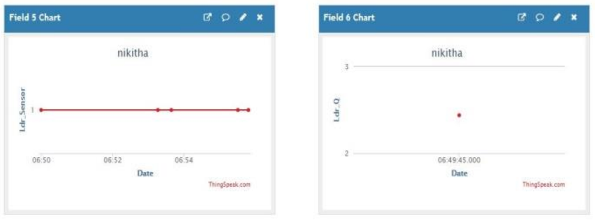


Fig 8: LDR sensor data in cloud

5. CONCLUSION

The virtual resource allocation in cloud-fog computing system is done using priority based Q Learning algorithm. As the service requests arrive based on the priority the service provider decides the action from its respective action space. Here the probability of the rejecting the requests is less compared to the conventional methods.

REFERENCE

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

[2]J. Liu *et al.*, "Application Partitioning algorithms in mobile cloud computing :Taxonomy ,review and future directions," *J. Netw.Comput. Appl.*,vol. 48, pp. 99–117, Feb. 2015.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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. 2398–2.