Adaptive Resource Management Utilizing Reinforcement Learning Technique in Inter-Cloud Environments

Vadla Pradeep Kumar 1* and Kolla Bhanu Prakash 2

1, 2 Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P, India

*Email: pradeepvadla@gmail.com

Abstract. In cloud computing, the cloud provider agent offers the quality of service (QoS) for different categories of cloud consumer agents. In general, the inter-cloud environment provides resources as a virtual machine (VM) instance representing processing power, Memory allocated in RAM, and secondary storage for consumer agents with QoS guarantees. A service level agreement framework with a reinforcement learning mechanism is considered for provisioning VM's for all categories of client classes. The parameters like cost of service, availability, and service demand are considered while provisioning VM's in the inter-cloud environment. QoS violation happens because another set of cloud consumer agents receives the less no of VMs. In our approach, the adaptive resource provisioning is integrated with reinforcement learning mechanism during the service admission process and ensures the collaborated cloud providers will gain more profits without violation of SLA.

Keywords: Resource provisioning, Reinforcement learning, Quality of Service, Service level agreement, Inter-cloud

1. Introduction

Cloud computing by its initial characteristics provides the feature of virtualizing [1] the resources at the level of computing, primary memory, and secondary memory requirements of cloud consumers. The factors which need to be measured and checked for violation are QoS (Quality of Service) parameters like response time, process time, and availability at the cloud provider’s side as mentioned in the contractual agreement. Resource management is a critical task in inter-cloud environments as multiple cloud providers cooperate to provide VM instances as resources. The key elements which need to be explored in inter-cloud environments are cloud consumer satisfaction, QoS violation, and adaptive management of VM instances. In the current work, a machine learning algorithm technique named reinforcement learning is incorporated in the admission control system while admitting the cloud consumers to request resources. Our model optimizes the profit of cloud providers while allocating the resources in the realization of VM instances and they need to ensure of not violating SLA (Service Level Agreements) by meeting the QoS demands of cloud consumers. In our approach blocking probability of constraints is applied while utilization of resources within SLA for measuring the price for cloud provider profit.

The major aims of the proposed reinforcement learning technique for resource provisioning are to develop a customer satisfaction scheme for supplying adequate VM’s for different classes of customers for the agreed QoS values. The rank of Service is an approach that is integrated with an admission control mechanism for computing blocking probability for providing quality service. Cloud provider’s gain is measured based on the utilization of their resources meeting the demands of cloud
consumers. The reinforcement learning technique for adaptive resource provisioning aims for attaining two major goals i.e. QoS of cloud consumers and cloud providers profit. The reinforcement learning technique uses a Markov choice mechanism to compute the profit of cloud providers while serving the demands of different categories of cloud consumer's for VM's. Mathematical simulation of the reinforcement approach was done by considering the QoS as a major concern while utilization of VM's of different cloud providers and tested for different cloud conditions. System models and related work are covered in the second section. The proposed reinforcement learning adaptation for resource provisioning is done in the third section. Performance Analysis and comparison of net gain profit for a different class of cloud consumers were discussed in the fourth section and finally, in the fifth section, the conclusion with future work details was highlighted.

2. System Model and Related Work

The cloud consumers request resources in the form of VM instances through any gadgets like desktop PC, laptops, or mobile phones. Figure 1 shows the proposed system model in which the cloud broker uses the reinforcement learning technique for provisioning VM resources to different categories of cloud consumers. The system model comprises cloud providers and Xc cloud consumers. The cloud provider has Num VM occasions that require for providing service to the different categories of cloud buyers. Cloud provider decides the amount of VM occasions Num for the classification c, important QoS for each class reliant on blocking probability of SLA and the award of sending acquired VM cases for kth class rc. Each cloud consumer of class k can request a set of VMs. Every customer of class k can ask for a list of VMs. The dvk represents the request vector for the required VMs for a particular category of cloud consumers of class k as shown in equation (1).

\[ dv_k = (vm_{i1}, vm_{i2}, ..., vm_{in}) \]  

(1)

where the value of vm is 1 or 0 denotes the VM instance is requested by cloud consumer is available or not. The award rej is given as below:

\[ rej = \sum_{i=1}^{n} pr^i * vm_i \]  

(2)

In equation (2) pr represents the price for ith VM instance paid by kth cloud consumer category. The boundaries conditions are changed after some time relating to the framework, for example, the outstanding task at hand, VM request, and the cloud provider server's maintenance cost. Thus, a cloud provider needs to change the expenses for the requested amount of VMs offered for each category when required. A cloud consumer can use the VMs if they agree to pay the specified amount for utilizing VM resources. The customer's solicitations landing pursue Poisson appropriation and every class of cloud consumer has entry rate \( \lambda_k \). The administration time \( \mu_k \) for each solicitation of kth cloud consumer class is assumed to be exponentially appropriated. The cloud provider needs to test of choosing the acknowledged solicitations. The cloud provider needs to consider cloud consumer interest for administration and Rank of Service \[23\]. The client request is firmly identified with the reward got from customers.

The contractual agreement of QoS between cloud providers and cloud consumers should always ensure not to get violated while provisioning the resources. Mie et al. \[2\] discussed the issue of a start to finish QoS ensures for Voice over IP (VoIP) \[19\] administrations is tended for violation or not. In Abrahao et.al. \[3\] the diverse internet administrations are facilitated in a common stage and provided to different categories of customers. The focal point was to deal with the limit of the common internet server farms to investigate the accessible assets to the supplier's best bit of leeway so a business objective is boosted. Ejarque et al. \[4\], Proposed a framework to encourage the asset of decreasing the expense of serving clients within specified QoS as decided by customers and the allotted assets of the customers are dependent on the data given by specialist co-operations. The assets are distributed to the customers as indicated by the business objectives \[20\] and customer's prerequisites \[21\].

Almeida et al. \[5\] propose another strategy that together tends to the asset assignment and affirmation control improvement issues in cloud situations. The proposed procedure considers the cloud provider's income, the expense of servers, and customers' prerequisites. Chen et al. \[6\], Proposed an approach for minimizing the QoS and cost while utilization of resources. Additionally, the creators recommend
a calculation to help cloud providers in overseeing server farms in a multi-cloud condition. Wu et al. [7] proposed a lot of calculations to limit the punishment cost and improving consumer loyalty levels by limiting QoS's imperative infringement [22]. The proposed conspire considers client profiles and suppliers' quality boundaries to deal with the dynamic idea of cloud conditions.

Dairo [8] proposed a scientific model is introduced to assess the presence of an IaaS in the cloud framework. A few exhibition measurements are proposed to investigate the conduct of a cloud server farm[18]. These measurements include accessibility, usage, and responsiveness. Alsarhanet. al. [9] utilizes a closeout model for asset distribution in inter-cloud with specific conditions. The main approach of the model works on providing the CP's benefit by making resources available through virtual resources. Prasad et al. [10] present a cloud asset acquisition approach which causes customers to choose a suitable cloud seller and executes unique valuing revenue-driven boost.

Hershey et al. [11] propose another plan for QoS checking the executives for an arrangement of frameworks (SoS) where any client from any area can share registering assets whenever. The proposed plot empowers QoS observing, the board, and reaction for big business frameworks that convey processing as assistance through a distributed computing condition. Jia et al. [12] proposed a versatile system for optimizing and maximizing the services of cloud providers. The structure is intended to improve the QoS of the delicate continuous mixed media applications in interactive media distributed computing. Hwang et al. [13] present new nonexclusive cloud execution models for assessing distinctive cloud administration models [17] and mashup or half breed mists.

3. Proposed RL Adaption for Resource Provisioning

The critical task of resource provisioning was managed here with the reinforcement learning (RL) technique to control the administration of multi-tasks based on the conditions to provide resources with the proposed model. RL model provides many benefits with improvement for the cloud provider in managing resource provisioning by meeting the requirements of maximizing profit without violation of QoS. The adaptations of the VM's resource provisioning are managed by implementing a strategy of admission policy [16] based on the request of categories of cloud consumers. This admission policy allows different categories of cloud consumers to send the request for different cloud providers to check for their QoS contract matching [24] without any cause of the violation. If the
consumer requests are getting satisfied by the cloud providers then they try to go for further requests to handle the situation to get maximum profit by provisioning VM instances. The handling of arrival of requests from different categories of cloud consumers uses the representation of handling of requests through Markov Decision Process (MDP) [25] to check for the likelihood of meeting different cloud providers QoS conditions for different cloud consumer requests. Q-learning [14] provides an alternate mechanism for the RL technique to enhance the utilization of MDP for different blocking probabilities by treating the requests for VM instances a state space. Table 1 represents the symbols used in this model for analysing the mathematical way of the parameters [26] for the proposed model.

Table 1. Parameters with Description

| Parameter | Description |
|-----------|-------------|
| Num       | No of VM's in the market |
| Xc        | No of cloud consumer |
| rej       | Award for renting VMs for jth category |
| Dvjk      | The demand vector of VM instances for a cloud consumer of kth category |
| Prijk     | The price for ith VM instance paid by jth category |
| λk        | kth category Inter arrival rate |
| μk        | kth category Inter service rate |
| St        | Total State space |
| Stk       | kth category of VM instances |
| Jc        | Categories of cloud consumers |
| Ωe        | Events recorded |
| ein       | Category-j Req event arrival |
| eti       | Category-j Req event departure |
| As        | Action work space |
| IR(S(t),e)(t), a) | The intermediate award function |
| πp        | Allocation policy |
| πp*       | Optimal allocation policy. |
| Dt        | The total time for request completion. |
| hv*       | Vector of differential award functions. |
| Exmlj     | The expected award of events probabilities |
| S(t+1)    | State space for next iteration |
| ARAD(j)   | A new request Reward for admitting of a class j. |
| DRDE(j)   | Cloud consumer Reward of departure of class j |
| λj        | The cloud consumers j acceptance rate of requests. |
| Prj       | Updated price. |
| Ar        | Max number of cloud consumers on new arrival |
| Ω         | Rate of decrease of the arrival rate as award r increases. |
| NG        | Cloud consumer Net gain of accepting j request. |
| Coj       | Cost of serving client j request. |
| NG*       | Optimal total net gain. |
| BPj       | Blocking probability of category j. |

3.1 Modelling Resource Allocation

The proposed framework changes as the QoS contract changes [27] from the request of categories of cloud consumers for a specific time interval. Depending on the instances of request changes for different time intervals by fixing probabilities of handling requests blocked. The cloud providers should finalize the decision of handling requests of variable VM instances of different categories of
cloud consumers for targeting the specific QoS contract for not a violation for targeted requests to get
a handle by the system.
In the currently proposed system, the handling of requests can show a discrete-time event structure.
The arrival of requests and handling of resource allocation is treated as probability computation. In the
RL model the states, categories, and target work n are analyzed by specifying equations.
The request of VM instances for each category of the state solution space of the cloud is given by
equation (3)
\[ S_t = \sum_{k=1}^{K} S_{t,k} \leq \text{Num} \]  
(3)
Where \( S_{t,k} \) is the number of required VM instances for \( k \)th category, and \( K \) is the set of cloud consumers’
categories.
Let \( \Omega \) denote the finite set of events in the system where:
\[ \Omega = \{ e_0, e_1, e_2, e_3, e_4, \ldots \} \]  
(4)
Where \( e_i \) denotes the request arrival event for class-\( i \), and \( e \) indicates the departure of class \( k \) request
(e.g. \( e = 1 \) means a request for class \( k \) departs the system while \( e_{i,k} = 0 \) means no request for class
\( k \) departs the system).
Let \( A_r \) mean the arrangement of permitted activities when the lease conditions for the recent
development are given. This set \( A_r \) can be defined as:
\[ A_r = \{ a_r : a_r \in \{0, 1\} \} \]  
(5)
where \( a_r = 0 \) denotes that a request is rejected, and \( a_r = 1 \) indicates that the cloud provider
accepts the request. Let \( \pi_p \) be a stationary policy that maps the current event and state of the
cloud environment to an action.
The planning methodology using approach \( \pi_p \) is an embedded constrained state Markov chain
progressing inconsistent time. Let \( e(t) \) be the event that occurs at time \( t \) and expect \( S(t) \) is the
state of the cloud condition at time \( t \) amid break \( [t-1, t] \). At that point, the middle of the
road remunerate for activity \( a \) at time \( t \) is registered as pursues:
\[ IR(S(t), e(t), a) = \begin{cases} r_{e_j, e'_j} \text{ if } e'_j(t) = e'_j, a_r = 1 \\ 0, e'_j(t) = e'_j, a_r = 0 \end{cases} \]  
(6)
Our objective is to find the optimal policy \( \pi_{p^*} \) that maximizes the average reward \( (\pi_{p^*}) \) such that
\( (\pi_{p^*}) \geq (\pi_p) \) and this can be expressed as follows:
\[ v(\pi_{p^*}) = \lim_{D_t \to p} \sum_{D_t=0} IR(S(t), e'_j(t), a_r) / D_t \]  
(7)
where \( D_t \) is the total time for request completion. For policy \( \pi \), any state can be reached by any
other state and the limit in (7) exists and is independent of the initial state.
The optimal policy \( \pi^* \) can be generated by solving Bellman’s equation for average reward:
\[ hv(S(t)) + v(\pi_p) = \max_{a_r \in A_r} \left\{ E_{e_j} \left[ IR(S(t), e'_j(t), a_r) + hv(S(t+1)) \right] \right\} \]  
(8)
where \( v(\pi_{p^*}) \) is the optimal reward, \( h \) is the vector of differential reward functions, \( \tau S(t) \) is the
average transition time corresponding to state-action pair, \( E_{e_j} \) is the expected reward over
the probability events, and \( S(t+1) \) is the next state which is a function of \( S(t) \), \( e_j(t) \), and \( a_r \).

3.2 Optimizing Resource Allocation in Inter-Cloud Environment
In this section, the reward function computation is analyzed for arrival customer’s requests for VM
instances using the RL model technique for a specific action. Let \( AR_{AD(j)} \) represent the reward for
admitting a new request of class \( j \). Then:
\[ AR_{AD(j)}(\pi_p, S(t)) = \max \{ \tau N_j + v(\pi_p, S(t) + e), v(\pi_p, S(t)) \} \]  
(9)
Assume \( DR_{DE(j)} \) models the departure of a class \( j \) request, which is defined as:
\[ DR_{DE(j)}(\pi_p, S(t)) = v(\pi_p, S(t) - e) \]  
(10)
Thus (7) could be re-written as:
The net gain of accepting the client \( j \) request is computed as follows:

\[
\text{Net gain} = \eta_j N_j - \text{Cost of serving a request of client } j.
\]

Where \( \eta_j \) is the cost of serving a request of client \( j \).

The reinforcement learning policy as in (13) can be expressed in terms of net gain where the CP attempts to maximize the net gain as follows:

\[
\text{NG} = \arg \max_{\eta_j N_j, \text{Cost of serving a request of client } j}\text{Net gain}
\]

Cloud provider utilizes reinforcement learning strategy to settle on choices that give high net gain addition. The reinforcement learning strategy must investigate all activities and favor the one that produces the most noteworthy increase. Along these lines, the learning procedure steadily favors activities that show up more commendable than others by evaluating an assortment of activities after some time. In our model, the reinforcement learning strategy picks the best activity that gives the most extreme increase. Thus, the learning procedure steadily favors activities that follow:

\[
\text{Begin for } t = 1 \text{ to } D_t \text{ do}
\]

\[
\text{End for}
\]

3.3 Algorithm for finding the optimal policy

**Input:** Number of VM’s, Number of classes, Number of clients, Blocked probability constraint for each class, Service price of each class (class1, class2, class3, class4). Constant parameters like \( \alpha \), \( \omega \). Time horizon (\( D_t \))

**Output:** new reward function after optimized

1. Arbitrarily initialize \( S(0) \in \mathbb{S} \), and \( \bar{v}_s \)
2. for \( t = 1 \) to \( D_t \) do
3. At state \( S(t) \), generate an event \( e(t) \)
4. \( \tau(t) = t - t - 1 \)
5. updates:
6. \( l_t = \text{IR}(S(t), e(t), \alpha) - \bar{v}_t \cdot \tau(t) + \hat{h}_{\text{ute}, t-1} (S(t)) - \hat{h}_{\text{ute}, t-1} (S(t - 1)) \)
7. \( u_t = u_{t-1} + \phi_{\text{ute}} \hat{h}_{\text{ute}, t-1} (S(t - 1)) \)
8. \( v_{\text{ute}} = v_{\text{ute}} + \alpha (\text{IR}(S(t - 1), e(t - 1), \alpha) - v_{\text{ute}} - \tau(t)) \)
9. \( ar = \arg \max_{\text{ute}} \{ \text{IR} (S(t), u_{\text{te}}(t), \alpha) + \hat{h}_{\text{ute}} (S(t + 1)) \} \)
10. Compute \( \text{IR}(S(t), e(t), \alpha) \)
4. Performance Evaluation and Results

In this section, the proposed model performance in the adaptation of resources provisioning was analysed for different QoS limitations and the reward function gets computed based on the satisfaction of VM instances request by considering all parameters mentioned in Table 1. The following are the key performance parameters which results from the proposed RL model are computing net gain for cloud provider by equation (14) and QoS are measured by the blocked probability of violation by cloud providers while allocated VM instances to cloud consumers by equation (15).

The system is simulated through python for a QWS dataset [15], available for inter-cloud environment with various QoS values and profits and tested on PC with configuration 4GB RAM, 1.3 GHZ Intel Core3 Processor. Figure 2 shows the results of increasing net gain of cloud providers for their different requests satisfied for different categories of cloud consumers. In this trial, we research how the QoS levels for customers influence the cloud provider increase. From Figure 3 we see that when the cloud provider gives an abnormal state of QoS regarding blocking likelihood its addition is debased essentially. As $BP_{i}$ diminishes, the QoS necessity for customers ends up stricter so that more demands ought to be shielded from dismissal. For this, client demands must be conceded all the more regularly to meet blocking likelihood limitations and the cloud provider can’t choose commendable demands. On the other hand, as $BP_{i}$ builds, the QoS prerequisites become less exacting so more demands can be rejected upon their entry and the cloud provider chooses the commendable demands as far as remuneration. A reinforcement learning-based plan should keep the blocking probabilities for all categories underneath the objective esteem paying little respect to the offered burden.

![Figure 2. Cloud Providers net gain under different system capacities](image-url)
Figure 3. Net gain from satisfying each category of cloud consumers under different Blocked probabilities

5. Conclusion and Future work
The resource provisioning without QoS violation is tested for a large set of VM instances request in inter-cloud environments. The Markov decision process provides a good strategy for cloud providers to boost performance in terms of profit generation. The reinforcement learning deals with gain computation which encourages cloud providers in adjusting the assets to meet the needs of cloud consumers to deal with QoS limitations. It is charming that when the QoS essentials become stricter, the cloud suppliers slant toward less advantage. On the other hand, when clients become less extreme for QoS, a cloud supplier can create more get. Future work is to test the reproduction approach in the ongoing cloud.

References
[1] Alsarhan A and Al-Khasawneh A Resource trading in cloud environments for utility maximisation using game theoretic modelling approach 2016. J. J. Parall. Emer. and Distr. Sys. 31319–333.
[2] Mei D and Meeuwissen H B Modelling end-to-end Quality-of-Service for transaction-based services in multidomain environments 2005 Proc. ITC19 1109–21.
[3] Abrahao B, Almeida V, Almeida J, Zhang A, Beyer D and Safai F Self-adaptive SLA-driven capacity management for Internet services 2006 Proc.NOMS557–568.
[4] Ejarque J, de Palol M, Goiri I, Juli’a F, Guitart J, Badia R and Torres J SLA-Driven semantically-enhanced dynamic resource allocator for virtualized service providers 2008 Proc eScience 8–15.
[5] Almeida J, Almeida V, Ardagna D, Cunha I, Franchelanci C, and Trubian M Joint admission control and resource allocation in virtualized servers 2010 J.Parall. Dist. Comp.70 344–362.
[6] Chen W, Cao J and Wan Y QoS-aware virtual machine scheduling for video streaming services in multi-cloud 2013 Tsi. Sci. Tech. 18308–317.
[7] Wu L, Garg SK, Versteeg S and Buyya R SLA-based resource provisioning for hosted software as a service applications in cloud computing environments 2013 IEEE Trans. Ser. Comp. 99 465–485.
[8] Dario B A stochastic model to investigate data center performance and QoS in IaaS cloud computing systems 2014 IEEE Tran. Parall.and Distr. Sys. 25560–569.
[9] Alsarhan A, Al-Sarayreh K, Ghuwairi A Al, and Kilani Y Resource trading in cloud environments for profit maximisation using an auction model 2014 Int.l J. Adva.Intell. Para. 6, 176-190.

[10] Prasad A C and Rao S A Mechanism Design Approach to Resource Procurement in Cloud Computing 2014 IEEE Tran. Com. 63 17-30.

[11] Hershey P C, Rao S, Silio C B and Narayan A System of systems for Quality-of-Service Observation and response in cloud computing environments 2015 IEEE Sys. J. 9 212-222.

[12] Jia G, Han G, Zhang D, Liu L, and Shu L An adaptive framework for improving quality of service in industrial systems 2015 IEEE Acc. 32129-39.

[13] Hwang K, Bai X, Shi Y, Li M ,Chen G and Wu Y Cloud performance modeling with benchmark evaluation of elastic scaling strategies 2016 IEEE Tran. Parall.and Distr. Sys. 27130-143.

[14] Watkins H and Dayan P TechnicalNote:Q-learning 1992 Mach. Learn.8 279-292.

[15] Al-Masri E, Mahmoud QH (2008) The qws dataset. http://www.uoguelph.ca/~qmahmoud/qws/indexhtml. Accessed: 12 Mar 2013.

[16] Sultanpure K A, Reddy L S S An energy aware resource utilization framework to control traffic in cloud network and overloads 2018 Inter. J. Elect. Comp. Eng. 81018-27.

[17] Sultanpure K A, Reddy L S S Job scheduling for energy efficiency using artificial bee colony through virtualization 2018. Inter. J. Intell. Eng. Sys. 11138-148.

[18] Lavanya K, Reddy L S S, Reddy B E Modeling of missing data imputation using additive LASSO regression model in microsoft azure 2018 J. Eng.App. Sci. 13 6324-34

[19] Garigipati N, Krishna R V A study on data security and query privacy in cloud 2019 Proc. Inter. Conf. Tre. Elect. Infor. ICOEI 337-341

[20] Praveen SP, Rao K T, Janakiramaia B Effective Allocation of Resources and Task Scheduling in Cloud Environment using Social Group Optimization 2018 Arab. J. Sci. Eng. 43 4265-72

[21] Praveen S P, Rao, K T An effective multi-faceted cost model for auto-scaling of servers in cloud 2019 Sma. Inn., Sys. Tech. 104 591-601

[22] Prakash K B Content extraction studies using total distance algorithm 2016 Proc. Int. Conf. App. Theo. Com. Comm.Tech( iCATccT) 673-679

[23] Prakash K.B Mining issues in traditional indian web documents 2015 Ind. J. Sci. Tech.8 172-176

[24] Prakash K B, Rajaraman A, Lakshmi M Complexities in developing multilingual online courses in the Indian context 2017 Proc.Inter. Conf. Big Data Analytics Compu. Intell. ICBDACI 339-342.

[25] Prakash K B, Rajaraman A, Perumal T, Kolla P Foundations to frontiers of big data analytics 2016 Proc. Intern. Conf. Cont. Comp.Infor. IC3I 242-247.

[26] Pradeep Kumar V, Kolla Bhanu Prakash QoS Aware Resource Provisioning in Federated Cloud and Analyzing Maximum Resource Utilization in Agent Based Model 2019 Inter. J. Innov. Tech.Exp. Eng. 8 2689 – 97.

[27] Pradeep Kumar V, Kolla Bhanu Prakash FLA-SLA Aware Cloud Collation Formation Using Fuzzy Preference Relationship Multi-Decision Approach for Federated Cloud2020 Pert. J.Sci.Tech.28 117-140.