A QoS and Profit Aware Cloud Confederation Model for IaaS Service Providers

Amit Kumar Das, Tamal Adhikary and
Md. Abdur Razzaque[*]
Green Networking Research Group
University of Dhaka, Dhaka-1000, Bangladesh
amit.csedu@gmail.com,
tamal.csedu@gmail.com,
razzaque@ieee.org

Eung Jun Cho, Choong Seon Hong[†]
School of Electronics and Information
Kyung Hee University Global Campus
1732, Deokyoungdaero, Giheung-gu, Yongin-si,
Gyeonggi-do, 446-701, South Korea
d2o2mask@khu.ac.kr,
cshong@khu.ac.kr

ABSTRACT

Cloud confederation is the union of cloud data centers that allows outsourcing user requests by a cloud provider during peak hours to liberate it from the constraint of limited physical resources. Federation extends some of the features of cloud like low cost, scalability, robustness and availability while increases the price and revenue of the providers along with maintaining a high QoS by effectively utilizing the resources of the data centers. Existing works on cloud federation do not consider all three categories of instances and hence cannot maximize their profits. Moreover, the process of establishing cloud federation and taking cost efficient decisions based on QoS parameters has not yet been provided in recent works. In this paper, we have designed a data communication model for the members of a cloud confederation, which exploits clear understanding of the usage pattern, types of requests in addition to infrastructure expenses. We develop an algorithm and a model for cost calculation, which enhances the decision making process over all the VM types (on-demand, reserved, spot) to increase resources utilization and profit. The simulation results, conducted on CloudSim, indicate that, our proposed model and algorithm enhances the profit, utilization, and QoS in a cloud confederation environment compared to a number of state-of-the-art approaches.

Categories and Subject Descriptors

H.4 [**Communications**]: Miscellaneous; D.2.8 [**Grid and cloud computing**]: Cost efficient—*Algorithms, performance measures*

General Terms

Grid and cloud computing, Algorithms

Keywords

Cloud Computing, Cloud Model, Cloud Confederation, Cloud VM types

1. INTRODUCTION

In recent days, Cloud Computing has turned out to be a consolidated archetype for delivery of services from beginning to end, on-demand provisioning of virtualized resources. By the appearance of this paradigm, along with sustainability of companies similar to Amazon, Microsoft, and IBM, the extensive envisioned dream of computing as a utility finally has moved towards reality. Currently consumers are capable to exploit resources and services in a pay-as-you-go approach from anywhere at anytime. Among the dissimilar methods to distribute Cloud services, Infrastructure as a Service (IaaS) allows Cloud providers to sell resources in the form of Virtual Machines (VMs) to consumers.

To be capable to suggest QoS guarantees without limiting the amount of accepted requests, providers must be able to dynamically enlarge the obtainable resources to serve requests. One probable basis for supplementary resources is hiring resources from other providers. In order to facilitate such circumstances, synchronization among providers has to be achieved, perhaps through founding of Cloud federation [1].

A Cloud federation allows providers to deal their resources by maintaining federation policies. In this archetype, providers aspire to conquer resource restriction in their home infrastructure, which can affect in satisfying customer needs, by outsourcing requests to additional members of the federation. In addition, Cloud federation allows underutilized providers to lease fraction of their resources to additional members of the federation, typically at cheaper prices, in order to evade killing their non storable compute capital. Both cases guide to improvement in revenue and elasticity for providers, if this chance is correctly used. By this we signify that, providers must create an intelligent conclusion regarding utilization of the federation depending on dissimilar conditions that they might face.

A difficult stipulation for providers occurs as they donate

---

[*] Dept. of Computer Science & Engineering, University of Dhaka, Dhaka 1000, Bangladesh

[†] Dept. of Computer Engineering, Kyung Hee University, Suwon, South Korea

IMCOM(ICUIMC)'14, January 9–11, 2014, Siem Reap, Cambodia

part of their ability in the form of spot VMs. Spot VMs are VMs that can be ended by providers whenever the current worth for running such VMs (distinct by the provider) exceeds the worth that the client is eager to disburse for using such resources, as in the case of Amazon EC2 spot instances [2]. This kind of VMs can be provided to users at a minor charge than on-demand VMs, typically in the spot marketplace, which works based on supplying in addition to demand. Existence of spot VMs surely reimburses IaaS Cloud providers, since spot VMs help them in creating revenue by escalating the operation of the data center even as waiting for inward on-demand requests. While a federated Cloud provider receives an on-demand request for VMs and there are no inactive/idle resources inside the data center, it has to choose either terminating spot VMs, or outsourcing the request to a different federation associate.

Pronouncement on outsourcing requirements or renting a part of idling resources to supplementary providers is a multifaceted problem that has been surveyed by numerous studies [3]. To the best of our knowledge, our work in this paper is the first attempt to work altogether with all the three types of VMs (On-demand, Reserved, Spot). Our key objective is to maximize provider's revenue, by satisfying SLA (Service Level Agreement). Our core contribution is proposing policies that assist making decisions when providers have diverse choices for all the three types of VMs about incoming requests, rejecting, outsourcing, or terminating.

The rest of the paper is organized as follows. The Section 2 describes some of the works related to our topics of interest. In section 3, cloud confederation model and assumptions have been described. In Section 4, the proposed mechanisms and cost analysis have been shown. The Section 5 presents the result of performance evaluation and simulation. In Section 6, we conclude the paper along with the future research directions.

2. RELATED WORKS

In spite of numerous newly proposed platforms for Cloud federation with dissimilar motivations in addition to incentives for parties to connect it [1], a lot of primary problems and questions regarding federation remain unanswered. One of these problems is deciding at what time providers ought to outsource their local requests to additional participants of the federation or how many and at what charge they ought to offer resources to the federation. The outsourcing difficulty is not measured only in the framework of federated clouds; it was also investigated as a means of rising capability or scalability of applications in hybrid Clouds [6], grid environment [7], and clusters [8].

In paper [3] the researchers present a profit-driven strategy for decisions correlated to outsourcing or selling idling resources. According to the authors, providers have the choice of shutting down idle nodes of the data center to save power. Though, they did not catch into account diverse types of VMs (e.g. on-demand and spot) in addition to probable actions like terminating low priority VMs.

A consumer satisfaction-oriented scheduling algorithm for serving requests was developed in [9]. Such an algorithm tries to exploit Cloud providers' revenue by accepting as many service requests as it can, as long as QoS is reserved at a certain level. In this view, contracting with additional service providers was taken into explanation as a technique to avoid rejection of consumer requests. One of the major differences between this and our approach is that we particularly focus on federation of IaaS providers that serve requests for VMs.

The trouble of how to value resources in addition to how price can impact of its use is not an unimportant one. Present public cloud service providers like Amazon, GoGrid and RackSpace frequently adopt unchanging pricing strategies for the infrastructure services they offer. Nevertheless, permanent pricing models are not appropriate for federated environments as a strategy to be functional among its participants, since it neither reflects present market value of resources due to dynamism in deliveign and demand nor generates some incentives for providers to connect the federation. Dynamic pricing of resources, though, lies outside the range of current work, it has been a topic of other studies [10].Thus, in this work, a strategy based on the provider operation, is applied by federated providers to dynamically charge resources.

The topic of leveraging spot VMs has just attracted significant concentration. In the paper [11], the authors have shown a probabilistic judgment representation to assist users decide how much to bid for a definite spot instance category in order to meet a positive monetary plan or a deadline. In paper [12], the researchers have developed a technique to decrease monetary expenses of computations by means of Amazon EC2 spot instances for resource provisioning. These mechanisms consider methods for rising customers' profit in using spot VMs, while we are interested in enhancing resource provisioning policies for providers in all the three kinds of VMs. Furthermore, the problem of dynamic distribution of data center resources to diverse spot market to exploit cloud provider's whole profits has been investigated in [4].

A small number of works consider the application of business-oriented policies in federated environment [13]. These policies typically encourage equality and ensure mutual remuneration for parties concerned in the federation. Study and growth of such techniques inspire both resource providers as well as resource consumers to connect and continue in the market. In paper [14], the researchers study the cooperative performance of numerous cloud providers and suggest a cooperative game technique. In [16], the authors mainly work on some policies to be applied by cloud IaaS resource providers to decide when to buy computational resources and how resources should be made available in the market for other IaaS providers. But they only consider on-demand and spot VM types in the approach. In [17], the authors also work on two different type of VMs called on-demand and reserved. Both of the papers didn't give any complete model collaborating with all the different type of VMs. Moreover, their concept of establishing cloud federation is based on global information server which can be a bottleneck and a single point of failure. On the other hand, our work is focused on detailed policies to be applied by cloud IaaS resource providers to decide a specific decision based on requirements and it can handle all the three types of VMs.

3. CLOUD CONFEDERATON MODEL AND ASSUMPTIONS

In our previous work [15], we have shown the architecture of cloud services and VM provisioning model. We have classified the requests into groups and used VM pool to provide VMs. Resource distribution and VM allocation to the requests are performed by a Cloud Resource Manager (CRM). Some of the existing cloud service providers support classification of VMs. For example, Amazon EC2 supports 18 different types of on-demand instances based on virtual core, compute unit, storage, clustering facility, number of I/O operation in unit time etc. However, Amazon instances are categorized into three categories based on pricing and resource provisioning policy: On-demand, Reserved and Spot instances. In this paper, we have considered these three VM allocation policies for data centers. The properties of these three categories are given below.

- *On-demand Instance*: On-Demand Instances allow pay for compute capability by the hour with no long-standing commitments. This reduces the expenses and complexities of scheduling, purchasing, and maintaining hardware as well as transforms what are usually large unchanging costs into much minor changeable costs. On-Demand Instances as well take away requirement of purchasing “safety net” capability to handle intermittent traffic spikes [2].
- *Reserved Instance*: Reserved Instances provide the choice to make a small, one-time payment for every instance we desire to reserve as well as in turn receive a major reduction on the hourly charge for those instances. The Reserved Instance Marketplace is also available, which provides the chance to advertise Reserved Instances if anyone want to need change [2].
- *Spot Instance*: Spot Instances permit clients to bid on unused Amazon EC2 capacity as well as run those instances for a period untill which their bid exceeds the existing Spot Price. The Spot Price changes occasionally based on supply and need, and clients whose bids meet or surpass it gain access to the accessible Spot Instances [2].

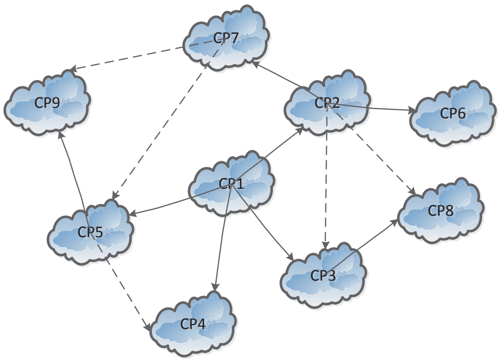

Figure 1: Arrangement and Interconncetion of CPs

The arrangement of the cloud providers (CP) and their interconnection is shown in Figure 1. In the figure, the connection for CP1 is shown by straight line and other connections not used by CP1, to set up its network, are given by dotted lines. Here, each CP maintains the connection of its neighbors and no centralized information or decision making server is present. The CPs communicate with each other through their internal link and make decision by themselves. The terms cloud provider, cloud service provider, cloud data centers and data centers are used interchangeably throughout the paper.

4. QOS & PROFIT AWARE CLOUD CONFEDERATION

In this section, a QoS and profit aware mechanism for VM allocation in a confederation environment is given. The procedure for communication among the members of confederation has been developed and described in details. Finally, the profit calculation for cloud providers has been shown using mathematical notations.

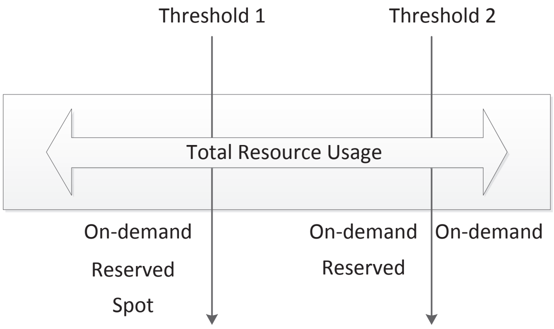

Figure 2: Resource usage threshold for various types of VMs

4.1 Policy for VM Allocation

To allocate VMs, we have considered two thresholds in terms of resource usage in the cloud data centers shown in Figure 2. When the resource usage is below Threshold1, all new requests of three VMs classes are allowed to enter the cloud data center and served. Between Threshold1 and Threshold2, no new spot instances are allowed. Existing spot instances are run and new reserved and on-demand instances are still allowed to enter the cloud data center. Above threshold2, all new reserved instances are rejected. Existing reserved and spot instances are allowed to run. Whenever new on-demand request arrives in this state, based on cost calculation one of two strategies can be undertaken. If the request can be served by another cloud service provider within reasonable price through cloud federation, help form neighboring cloud providers is taken. If the cost of service through cloud federation is higher and service can be provided by terminating some spot instances, those spot instances are terminated for giving scope of running the higher-priority on-demand request. If the request cannot be served by stopping the spot instances, the request is rejected and no SLA agreement is done.

4.2 Proposed Algorithm for Cloud Federation

The arrangement of cloud providers in a cloud federation is given in Figure 1. When new on-demand request arrives

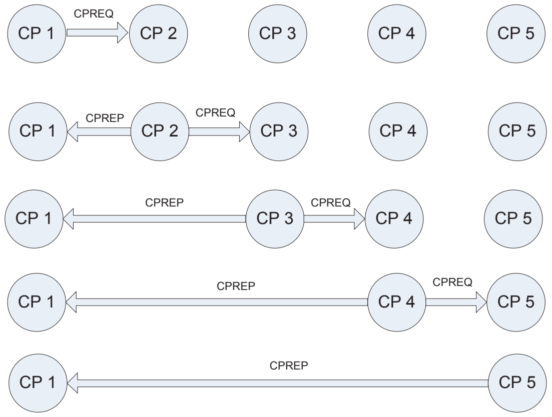

Figure 3: Message passing among CPs

or requirement of an existing on-demand request increases that cannot be satisfied by the cloud provider, it asks its neighboring cloud data centers for their service prices. No centralized data directory is used since it can be a bottleneck and failure of the centralized server will cease the whole cloud federation policy. From the list of the service prices of the neighbors, one is selected by considering its price for service (S_{cost}), time for data transmission (DT_{time}) and availability of similar type of VM (VM_{avail}).

If VM of same time required by the request is available in the remote service provider, time for creating new VM (VM_{create}) will be saved. As a result, SLA violation can be avoided. Forethought will have to be ensured such that,

$DT_{time} + S_{time} < SLA_{time}$,

if time condition is present in SLA. DT_{time} should not be long enough compared to VM_{create}. The value of VM_{avail} for a cloud provider should be 1 or 0 depending on the absence or presence of similar type of VMs in its VM pool respectively. If VM of same type is not present but enough resource for creating new VM is present, the value of VM_{avail} will be 0.5. Based on the above properties, service log (SL) for neighboring cloud provider is created using the following equation,

$$SL_i = \alpha \times S_{cost_i} + \beta \times DT_{time_i} + \gamma \times VM_{avail_i}, \qquad (1)$$

where, SL_i is the service log for the neighboring service provider i. The value of α, β, γ should be assigned in such a way that, $\alpha > \beta > \gamma$ and $\alpha + \beta + \gamma = 1$. Based on the calculated service log, the cloud provider which has the minimum SL value is selected. The algorithm for VM allocation in a cloud federation enabled data center is shown in Algorithm 1. The algorithm is run every time when new request arrives at the data center for getting service.

4.3 Cloud Federation Information Transfer Methodology

In Figure 3, we have seen 5 Cloud Providers (CP) naming CP1-CP5. Now CP1 will create flooding for checking available resources for serving the requests to its confederation cloud. CP1 will send a CPREQ message which contains the “Request Type (RT)”, to its neighboring CPs. When CP2 receives a CPREQ message from CP1, its CRM will check whether it has any free space and same type of virtual machine (SVM) or not. If CP2 can satisfy both of the conditions, It simply send a CPREP message to CP1, containing FS=1 (FS=free space, means it has enough space to serve the request) and SVM =1. If CP2 meets only the first condition that it has enough free space to serve the request but don't have any SVM, the CPREP will contain FS=1 and SVM=0. If it doesn't have enough free space, CPREP message will contain FS=0 and SVM=0. It will then forward the CPREQ message to its neighboring CPs. The CPREQ message will contain a request number, CP number to avoid duplicate transmission, hop count to limit the number of message transmission and last CP number which will guide to send the CPREP message. The CPREP message will contain requesting CP number who has sent the CPREQ message, the CP number of the data center which is responding and the cost for the asking VM or request type. For each transmission, hop count value of CPPEQ will be decreased by 1 and when hop count reaches 0, the message will not be transmitted any further. Each data center will maintain a table containing two fields; destination

Algorithm 1 *VM Allocation in Cloud Federation*

INPUT: *RU:Resouce Usage*
NRS: Size of New Request
RT: Request Type
TR: Total Resource
OUTPUT: *Cost effective VM allocation*

```
while New request arrives do
  if RT='spot' then
    if RU< threshold 1 then
      Allocate VM
    else
      Reject request
    end if
  end if
  if RT='reserved' then
    if RU< threshold 2 then
      Allocate VM
    else
      Reject request
    end if
  end if
  if RT='on-demand' then
    if RU< threshold 2 && RU+NRS<TR then
      Allocate VM
    else
      for each each neighbor i do
        Calculate $SL_i$
      end for
      if $SL_i \neq \varnothing$ then
        Choose CP based on minimum $SL_i$
      else
        Reject request
      end if
    end if
  end if
end while
```

Table 1: Content of CPREQ Message

CPREQ Message
CP (Cloud Provider) Number
Request Number
Request Type (RT)
Hop Count

Table 2: Content of CPREP Message

CPREP Message
REQ CP (Cloud Provider) Number
CP Number
FS (Free Space)
SVM (Same type of Virtual Machine)
Cost

and next hop. Whenever a CPREQ message is received, a data center will fill up the destination field from the CPREQ message and the next hop field from the last CP number of the CPREQ message. From the time difference between the CPREQ and CPREP, the DT_{time} is calculated. The contents of the CPREQ and CPREP message are shown in Table 1 and Table 2 respectively.

4.4 Cost and Profit Calculation

The overall strategy of cloud federation is profit based and it has been incorporated to maximize the profit of a cloud data center. Let N_s, N_r, N_o, N_{out} and N_{serv} be the number of spot, reserved, on-demand, outsourcing and cloud federation serving instances respectively. In a confederation environment, the total profit, P_{total} at any instant of t can be given as,

$$P_{total}(t) = P_s(t) + P_r(t) + P_o(t) + P_{out}(t) + P_{serv}(t), \quad (2)$$

where, $P_s(t), P_r(t)$ and $P_o(t)$ are the profits from spot, reserved and on-demand instances running in that data center at that instant of time t. $P_{out}(t)$ is the profit form outsourcing and $P_{serv}(t)$ is the profit from serving requests of other cloud providers. The cost for outsourcing should be less than $P_{out}(t)$ and for a particular instance, serving it in cloud federation will make less profit than serving it as an on-demand instance. That is, $P_{serv}(t) < P_o(t)$ to make the cloud federation profitable for both the local and remote cloud service providers. $P_s(t)$ can be calculated by summing the profit from all the spot instances.

$$P_s(t) = \sum_{i=1}^{N_s} R_{spot_i}(t), \quad (3)$$

where, $R_{spot_i}(t)$ is the revenue from the spot instance i at time t. Similarly the profit for the reserved and on-demand instances can be gicen by,

$$P_r(t) = \sum_{i=1}^{N_r} RU_{r_i}(t) \times R_{reserved}, \quad (4)$$

$$P_o(t) = \sum_{i=1}^{N_o} RU_{o_i}(t) \times R_{on-demand}, \quad (5)$$

where, $RU_{r_i}(t)$ is the units of resource utilized by a reserved instance i and $R_{reserved}$ is the rate of each unit of reserved instance respectively. Similarly, $RU_{o_i}(t)$ is the units of resource utilized by an on-demand instance i and $R_{on-demand}$ is the rate of each unit of on-demand instances. Profit from the outsourcing requests, P_{out} for any time instance t is give by,

$$P_{out}(t) = \sum_{i=1}^{N_out} R_{out_i}(t). \quad (6)$$

The profit from serving requests from other CPs, $P_{serv}(t)$ can be given by,

$$P_{serv}(t) = N_{serv} \times R_{serv}(t), \quad (7)$$

where, R_{serv} is the revenue of serving each of the requests in cloud federation. Putting it all together, the overall equation for profit calculation can be given by,

$$\begin{aligned} P_{total}(t) &= \sum_{i=1}^{N_s} R_{spot_i}(t) + \sum_{i=1}^{N_r} RU_{r_i}(t) \times R_{reserved} \\ &+ \sum_{i=1}^{N_o} RU_{o_i}(t) \times R_{on-demand} + \sum_{i=1}^{N_out} R_{out_i}(t) \\ &+ N_{serv} \times R_{serv}(t). \quad (8) \end{aligned}$$

Our goal is to maximize P_{total}. If outsourcing is not required, P_{total} itself is maximized. When outsourcing is required, P_{total} can be maximized by trading off among the $P_s(t), P_{out}(t)$ and $P_o(t)$. If number of spot instances is reduced, more number of on-demand instances can be served and number of requests outsourced will also be reduced. Therefore, $P_s(t)$ and $P_{out}(t)$ will decrease and $P_o(t)$ will increase in such a way that, in the next time stamp t′,

$$P_o(t') - P_o(t) >= (P_s(t) - P_s(t'))(P_{out}(t) - P_{out}(t')).$$

Therefore, at any instance of time, if the servicing of on-demand instances by terminating of spot instances makes more profit for the next time stamp, on-demand requests are not outsourced. Otherwise, outsourcing is performed.

5. PERFORMANCE EVALUATION

To perform the evaluation of our proposed methodology, first we have set up a simulation environment. The results shown here are collected from experiments performed in CloudSim, a distributed simulation environment toolkit [18]. The simulation was performed for IaaS cloud providing VM, storage, bandwidth, etc.

Throughout the simulation, we have considered VMs of five different sizes. The configuration mostly resembles the Amazon EC2 Micro instances (613 MB of memory, up to 2ECU), M1 Small instance (1.7 GB of memory, 1 EC2 compute unit), M1 medium instance (3.75 GB memory, 2 EC2 compute unit), M1 large instance (7.5 GB memory, 4 EC2 compute unit) and High CPU Medium Instance (1.7 GB memory, 3.25 EC2 Compute unit). The cost of the VMs is considered to be similar to Amazon AWS pricing model. For spot instances, we assume that the bidding for an instance will not be exceeded the price of a reserved or on-demand instance, since it will not be beneficial to run a spot instance with a cost higher than the on-demand instances.

The whole environment is established containing 12 data centers. The OPEX for all the data centers is the same because of having same resources and almost similar amount of user operations. Each of the data centers contains 32 physical servers. The servers contain a quad-core processor, 8 GB of RAM and 1TB of local storage. The data centers are connected by 8 Mbps transmission link. The request arrival rate at the data centers is considered as 64 requests per hour.

The simulation was performed on our proposed QoS and Profit Aware Cloud Confederation Model (QPC), Resource Provisioning in a Federated Cloud (RFC) [16], VM Provisioning Method to Improve Profit and SLA Violation (VPM) [17]. The comparative study among these three policies is shown in the following graphs. The experiments are performed multiple times to achieve accuracy and the average of the results is shown as the ultimate outcome here.

Figure 4 shows the experiment result of serving time for different number of requests served. The figure shows that, our proposed QPC takes less time to serve same number of requests compared to RFC and VPM. This is because, we have used the policy of recycling created VMs from the VM pool. As a result, time for VM creation can be saved and more requests can be served at that time. None of RFC and VPM used the idea of VM pooling and hence takes larger time to create and serve VMs.

Profit for different policies has been calculated by using the Amazon AWS billing policy. The profit is calculated several times and normalized value has been given here. In Figure 5, the profit for different cloud federation policies has been shown. The results show that, since we incorporate all three VM classes and gain more profit from outsourcing compared to others, our proposed QPC has achieved maximum profit compared to others.

In Figure 6, number of rejected requests for varying hop count is given. As the number of hop count increases, the number of servers also increases. So the scope of outsourcing also increases. As a result, more number of requests which had to be rejected otherwise can now be outsourced and it increases the total profit of the cloud service provider.

In Figure 7, resource utilization for varying request arrival rate has been shown. The figure shows that, as the rate of arriving request increases, the total resource utilization in the data center also increases. However, due to designing two thresholds and controlling outsourcing policies effectively, resource utilization of our proposed QPC is greater than others and hence more requests can be served while less number of requests need to be rejected.

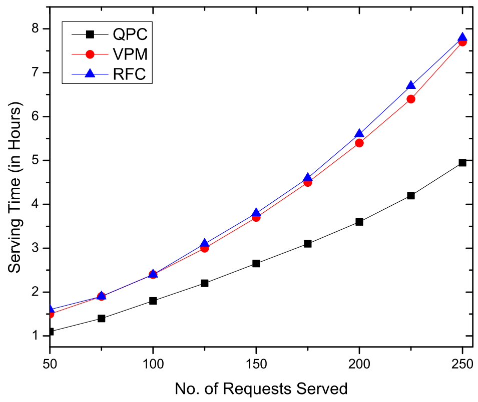

Figure 4: Serving time for varying no. of requests

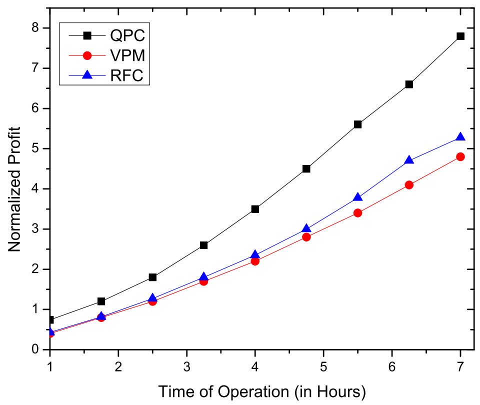

Figure 5: Profit for varying time of operations

6. CONCLUSIONS

In this paper, we have shown a cloud confederation model along with algorithm to increase IaaS providers' revenue when the provider is a member of a cloud confederation. Since every provider has limited amount of capability, increase in load possibly will overload a provider's data center as well as may result in QoS infringement or users' request rejection. Providers that support dissimilar types of QoS and pricing plan for VMs, have the option of canceling their less profitable VMs (spot VMs) in favor of additional profitable requests (on-demand VMs). Providers can also get advantage from federation through outsourcing requests to additional members of the federation by means of least load.

Comparative study in performance evaluation section shows that our proposed cloud confederation policy outperforms the existing works. Our proposed methodology maximizes profit of cloud data centers while increase resource utiliza-

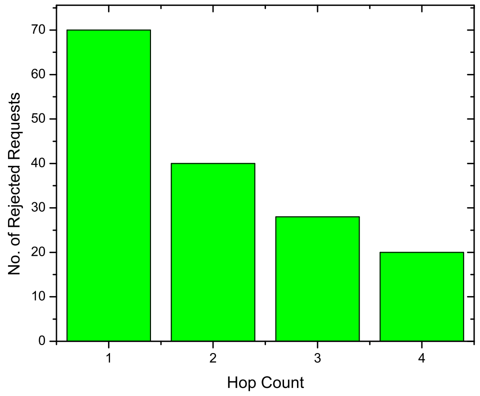

Figure 6: No. of rejected requests for varying hop counts

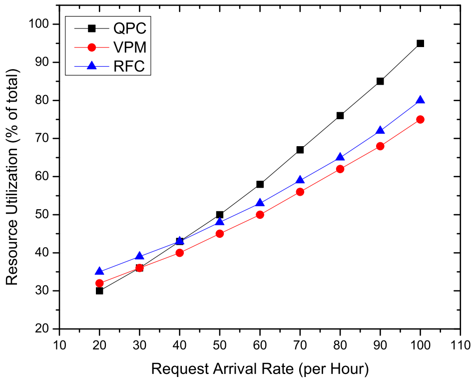

Figure 7: Resource utilization for varying No. of requests served

tion through serving huge number of requests.

We are planning to investigate some approaches where judicious decissions can be taken to terminate spot VMs for increasing providers revenue. Cloud resource allocation and load balancing among the data centers is also one of our topic of interest.

Acknowledgment

This research was jointly supported by Information Society Innovation Fund (ISIF) Asia Grants 2013 in favour of Driver Distraction Management System Using Sensor Data Cloud, and by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2010-0020728). Dr. CS Hong is the corresponding author.

7. REFERENCES

[1] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-oriented federation of Cloud computing environments for scaling of application services," in *Proc. of the 10th Int'l Conf. on Algorithms and Architectures for Parallel Processing (ICA3PP)*, Springer, May 2010, pp. 13-31.

[2] Amazon Elastic Compute Cloud (Amazon EC2), *http://aws.amazon.com/ec2*, accessed on 10 August 2013.

[3] I. Goiri, J. Guitart, and J. Torres, "Characterizing Cloud federation for enhancing providersï£¡ profit," in *Proc. of the 3rd Int'l Conf. on Cloud Computing*, IEEE Computer Society, Jul. 2010, pp. 123-130.

[4] Q. Zhang, E. Gurses, R. Boutaba and J. Xiao, "Dynamic resource allocation for spot markets in Clouds," in *Proc of the 2nd Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE)*, Mar. 2011.

[5] L. Rodero-Merino, L. M. Vaquero, V. Gil, F. Galan, J. Fontan, R. S. Montero, and I. M. Llorente, "From infrastructure delivery to service management in Clouds," *Future Generation Computer Systems*, vol. 26, no. 8, pp. 1226-1240, Oct. 2010.

[6] R. V. den Bossche, K. Vanmechelen, and J. Broeckhove, "Cost-optimal scheduling in Hybrid IaaS Clouds for deadline constrained workloads,"*IEEE Int'l Conf. on Cloud Computing*, Los Alamitos, USA: IEEE Computer Society, 2010, pp. 228-235.

[7] H. Kim, Y. el Khamra, S. Jha, and M. Parashar, "Exploring application and infrastructure adaptation on hybrid Grid-Cloud infrastructure," in *Proc. of the 19th ACM Int'l Symposium on High Performance Distributed Computing*, Chicago: ACM, June 2010, pp. 402-412.

[8] M. D. de Assunc, A. di Costanzo, and R. Buyya, "A cost-benefit analysis of using Cloud computing to extend the capacity of clusters," in *Cluster Computing*,vol. 13, no. 3, pp. 335-347, Sep. 2010.

[9] Y. Lee, C. Wang, J. Taheri, A. Zomaya, and B. Zhou, "On the effect of using third-party Clouds for maximizing profit," in *Algorithms and Architectures for Parallel Processing*, Springer Berlin Heidelberg, 2010, vol. 6081, pp. 381-390.

[10] M. Mihailescu and Y. M. Teo, "Dynamic resource pricing on federated Clouds," in *IEEE Int'l Symposium on Cluster Computing and the Grid*, 2010, pp. 513-517.

[11] A. Andrzejak, D. Kondo, and S. Yi, "Decision model for Cloud computing under SLA constraints", *Proc. of the IEEE Int'l Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS)* Aug. 2010, pp. 257-266.

[12] S. Yi, D. Kondo, and A. Andrzejak, "Reducing costs of Spot instances via checkpointing in the Amazon Elastic Compute Cloud," in *In Proc. of the 2010 IEEE 3rd Int'l Conf. on Cloud Computing (Cloud)*, Washington, DC, USA: IEEE, 2010, pp. 236-243.

[13] E. Gomes, Q. Vo, and R. Kowalczyk, "Pure exchange markets for resource sharing in federated Clouds," in *Concurrency and Computation: Practice and Experience*, vol. 23, 2011.

[14] D. Niyato, A. V. Vasilakos, and Z. Kun, "Resource and revenue sharing with coalition formation of Cloud providers: Game theoretic approach," in *Proc. of IEEE CCGrid*, IEEE, May 2011, pp. 215-224.

[15] A. K. Das, T. Adhikary, Md. A. Razzaque, C. S. Hong, "An Intelligent Approach for Virtual Machine and QoS Provisioning in Cloud Computing," in *Int'l Conf. on Information Networking (ICOIN), Bangkok, Thailand*, 27-30 january, 2013.

[16] A. N. Toosi, R. N. Calheiros, R. K. Thulasiram, R. Buyya, "Resource Provisioning Policies to Increase IaaS Providerï£¡s Profit in a Federated Cloud Environment," in *High Performance Computing and Communications (HPCC)*, 2011.

[17] P. Komal Singh and A.K Sarje, "VM Provisioning Method to Improve the Profit and SLA Violation of Cloud Service Providers", in *Cloud Computing in Emerging Markets (CCEM), 2012 IEEE Int'al Conf.* 2012, Bangalore, India.

[18] R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", in *Software: Practice and Experience, vol. 41, no. 1, pp. 23-50*, 2011.