Maximizing SLA and QoE in Heterogeneous Cloud Computing Environment

Md Sabbir Hasan¹, Eui-Nam Huh²
¹Department of Computer Engineering, Kyung Hee University, Suwon, South Korea
²Department of Computer Engineering, Kyung Hee University, Suwon, South Korea

Abstract - Cloud Computing delivers users a proficient way to dynamically allocate computing resources to meet demands. The use of Server virtualization techniques for Cloud Computing platforms provide great elasticity with the capability to consolidate several virtual machines on the same physical server, to resize a virtual machine capacity and to migrate virtual machine across physical servers. A key challenge for Cloud Service providers is proper resource management while taking into account of Service Level Agreement and Quality of Experience of users. Reserving resources would be beneficial to improve the quality of service, if the actual demand of the user is known in advance. However, in reality, the actual computing demand can be pragmatic only at the point of actual usage. In this paper, we propose a demand prediction method constructed on present usage of resources, aiming better provisioning of resource demand. Our proposed VM allocation is mapped into the multidimensional bin-packing problem, which is NP-complete. To solve this problem, we have designed heuristics for quantitatively optimizing the VM allocation. The simulation results show that our scheme performs better comparing to the existing VM allocation schemes in cloud computing environment, in terms of maintaining SLA and better QoE.

Keywords: Quality of Experience, Service Level Agreement, Cloud Computing, VM Resource Allocation.

1 Introduction

Cloud Computing has received significant attention recent times as the hype is created and responded largely by the companies like Amazon, IBM, Google, Yahoo!, Microsoft, Sun, NASA and RackSpace by making their own cloud platforms for consumers and enterprises to access the cloud resources through services. With the rapid development of Virtualization technology including the advantage of isolation, consolidation and multiplexing of resources, became key role to deploy in modern data centers [1]. Due to virtualization, numerous tasks are seen as a single entity in a virtual machine. Since Infrastructure-as-a-Service (IaaS) is a computational service model applied in the cloud computing paradigm, Virtualization technologies are used these days to support computing resource access by the users in this model. Users can specify required software stack such as operating systems, software libraries, and applications; then package them all together into virtual machines (VMs). Finally, VMs will be hosted in a computing environment operated by cloud providers. [2].

QoS requirement can be formalized in Service Level Agreement that serves as the foundation for the expected level of service between the Cloud consumer and the Service or Cloud provider. However, Infrastructure providers often end up over provisioning of resources to maximize the Service Level Agreement that results poor Resource Management and Large Operational cost. Contrary to that, under provisioning of resources will increase SLA violation and affect QoE of Users. Quality of experience (QoE), sometimes also known as Quality of Service, is a subjective measure of a user's experiences with a service in Cloud Computing Environment. The poor QoE will dissatisfy the User. A typical user-related measure is the mean opinion score (MOS), which can be determined from subjective analysis, usually used to measure the Quality of Experience of the User's. Nevertheless, Subjective analysis is not enough while predicting the resource demand of User that might require during execution time. So we build an exponential relationship between SLA violation and QoE, based on Subjective and Quantitative analysis to maintain better Service Level Agreement.

However, the research in SLA and QoE in Cloud Computing Environment is still in its infancy, and several technical issues remain open. One important issue is to meet the QoS requirements of Cloud services that require a different quantity of VM resources at run-time [3]. Inappropriate VM resource allocation in this environment may result in resource waste and Service quality degradation.

Nevertheless, the VM resource allocation is challenging due to the dynamic nature of the workload and cloud platform. Specifically, this is demanding in terms of the VM resource requirement, multimedia service heterogeneity and the heterogeneous network conditions [3], [4], [5]. It is cumbersome for a cloud provider to perform over commitment of VM resources for processing media service tasks, which may have different QoS requirements and unpredictable resource consumption. Moreover, the irregular
spikes and bursts of user activities in interactive cloud-based applications complicates the over commitment estimation of VM resources.

In this paper, we tackle the aforementioned challenges of VM resource allocation in heterogeneous cloud environment. We propose a VM resource allocation model that optimally allocates VM resources to a set of physical machines/servers by considering the dynamic VM resource requirements of cloud multimedia services. It ensures the minimum SLA Violation, while maintaining high system utilization by avoiding over provisioning the VM resources to the services. The proposed VM resource allocation model is designed to:

- Minimize the number of physical servers/machines for energy savings;
- Reduce SLA violation to improve QoE;
- Achieve load balancing or overall utilization of physical resources.

Several experiments are carried out to validate the efficiency of our proposed VM resource allocation model in heterogeneous cloud platform. These experiments are conducted for different patterns of workload for various environments. We also have compared our proposed algorithm with three other existing algorithms in cloud platform, which comprised of LR-MMT [6], BFD, FFD [7]. The results include the performance of SLA Violation reduction and better QoE of Users.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 describes our VM resource allocation model and heuristics for the current multidimensional bin packing problem. Section 4 presents experimental results and performance comparisons. Finally, Section 5 concludes the paper.

2 Related Works

Our work is based on Dynamic Consolidation of Virtual Machine’s retaining strict Service Level Agreement. There are several research groups in both academia and industry, working on Energy aware resource allocation and management by performing static and dynamic consolidation of VM’s and servers.

Cardosa et al. [8] proposed a solution for VM placement and power-efficient consolidation of VMs in modern data centers where it runs heterogeneous applications. They have adopted min, max, share parameters of XEN and VMWare that represents the Utilization limit of upper and lower of CPU allocation and sharing same resources by different VMs. They also considered a priority based approach for peak load of enterprise environment. As a result it does not support strict SLA and VM allocation is static. Verma et al [9] described a power aware application placement framework in which at each time frame the placement of VMs is optimized to minimize the power consumption and maximize the performance at certain level. The main difference with his work is that, our proposed algorithm doesn’t violate Strict SLA requirement when workload is varied and unpredictable.

Stilwell et al [10] proposed a formulation of the resource allocation problem in shared hosting platform for static workloads with servers that provide multiple types of resources. Their algorithm runs faster in large systems and fulfill QoS requirement but it lack dynamicity when workload is varied and unpredictable. Like him other researchers [11], [12] also studied VM resource management techniques to maintain QoS requirement when workload is static in Cloud Computing. Wood et al [13] developed the Sandpiper System that monitors and detects hotspots and reconfigure VMs when is necessary. In order to choose which VMs to migrate , their system sorts them using volume-size-ratio, which is a metric based on CPU, network and memory loads where as we considered both size of VM and the migration time required to maintain strict SLA.

In contrast to above approaches, Our algorithms that includes Host Utilization, power consumption increment of Hosts and migration time of VMs and further modification from their approach shows better result in terms of SLA Violation and QoE of Users. Our Estimator for Host overload detection provides better consolidation of VMs as it reduces the number of unnecessary VM Migration.

3 Problem Statement

As depicted in figure (1) the system model of Cloud Computing environment consists of three components, i.e., user, Cloud Broker and Cloud Provider. Each Cloud Provider supplies a pool of resources to the User. Cloud Broker works as a Centralized Unit or 3rd party which interacts with both Users and Cloud Providers having some core functionalities. Our approach in this paper is to predict the resource demand based on the current usage of resources by Cloud Providers and their workload of the servers, later allocate VM based on the actual demand to improve SLA and QoE of Users. So, we ignore to describe other functionalities of Cloud Broker. We propose the VM resource Allocation model as Multidimensional Bin Packing Problem [14], which is NP complete. The target is to find the minimum number of Physical Host to place the VM’s, with respect to Physical Host’s capacity. To solve this problem, we have designed a linear Programming (LP) model, as well as heuristics to quantitatively optimize the VM allocation. The VM resource allocation process can be static or dynamic. In static VM allocation, VM capacities are configured using peak load demands of each workload. The utilization of the peak load demand ensures that the VM does not overload and stay in the same physical servers during their entire lifetime. However, it leads to idleness due to the variable VM resource demand. In dynamic VM allocation, VM capacities are configured dynamically according to the current media workload.
demands. However, it may require migrating VMs between physical servers in order to: (i) pull out physical servers from an overloaded state when the sum of VMs capacities mapped to a physical server becomes higher than its capacity; (ii) turn off a physical server when the VMs mapped to it can be moved to other physical servers. In our scenario, we allow the virtual machine capacities to be varied on demand. We also introduce a VM migration policy to overcome from overloading the host, resulting better Quality of Service to User’s.

![Cloud Computing Environment](image)

**Figure (1) Cloud Computing Environment**

### 3.1 Resource Demand Prediction

Predicting the actual VM demand greatly minimizes the VM allocation problem that results less energy consumption and service downtime. However, the demand cannot be known in advance. Therefore, present usage pattern of all the cloud providers is used to determine the future VM demand and provide on demand resource addition if needed in our approach. The Broker is responsible for accumulating the information as a centralized entity. The usage history might unravel some patterns that might help to predict the actual usage amount. For our experiments, we use different pattern of workload to create uncertainty predicting the actual demand, but the functions we considered to measure the resource demand, improves resource utilization and performance. So we formulate, resource required to satisfy present demand, $R_i \geq \{Q(n), W_{avg}, SLA \}$. We carefully choose the functions to get better likelihood depending on the present usage of resources.

**Definition 1(Service Queue Length):** $Q_m(t)$ denotes the number of class-$m$ request arrive at the time instant $t$, $t \in \{1, 2, 3..T\}$ and $m \in \{1, 2, 3..k\}$. Every class-$m$ request has deadline $d_m$. So if a class- $m$ request arrives at $t$ time that must be served no later than $\{t + d_m\}$. We consider the arrival rate is Poisson process. So the arrival rate of incoming request is $\lambda$ and mean duration of served request is $\frac{1}{\lambda}$. So for class-$m$ request, arrival service request will be $\lambda_m$ and that should be $\lambda_m < \lambda$. Using little’s law, average number of request in the queue $Q_m(t) = \text{Arrival rate} \times \text{time to serve request}$. For Multimedia application (Instant Channel Change in IPTV), where deadline $d = 0$, each request must be served at the instant time. So number of servers needed at time $t$ will be $Q_I$. If we have number of servers at the Cloud provider side as $(S_1, S_2, S_3......S_T)$, then to satisfy the entire request at time instant $t$, we have to have $S_i \geq Q_i$.

So for a time frame from $t_0$ to $t_0 + t_n$:

$$\sum_{n=t_0}^{t_0+t_n} S_n \geq \sum_{n=t_0}^{t_0+t_n} Q(n) : \forall t_0, t_n \in t$$  \hspace{1cm} (1)$$

For a strict condition:

$$\sum_{n=t_0}^{t_0+t_n} S_n = \sum_{n=t_0}^{t_0+t_n} Q(n) : \forall t_0, t_n \in t$$  \hspace{1cm} (2)$$

But that would miss some of the deadlines that have been requested.

**Definition 2 (Server Workload):** In heterogeneous cloud environment, servers workload fluctuates depending on the behavior of different applications resource requirements. We consider, the average arrival rate of lock to a data item is $\lambda_l$ and locks the data item for $t_h$. Then the contention to that data item appears $C = \lambda_l t_h$. The probability of application contention in the server can be formulated $P_{cont} = \frac{1}{d} \lambda_l t_h$, where $0 \leq d \leq 1$. So average workload for Application on the server:

$$W_{avg} = \frac{1}{t_n - t_0} \left( \frac{1}{d} \lambda_l t_h \right) ; \quad 0 \leq d \leq 1$$  \hspace{1cm} (3)$$

For all the servers from different Cloud provider’s workload can be statistically measured by using the formula for $t_0$ to $t_n$ timeframe.
Definition 3 (SLA Violation rate): Maintaining Strong Service Level Agreement (SLA) and meeting QoS requirement is enormously important for Cloud Computing Environment. Overprovisioning of resources to meet target SLA can increase the power consumption whereas Energy-aware resource management is highly important these days. Basically QoS requirements can be governed by service response time or throughput, and that can be varied for different application. However for Cloud providers, workload independent metrics should be considered to calculate SLA percentage or violation. For measuring SLA violation in IaaS environment, we considered two metrics: (1) the percentage of time an active Host have experienced 100% CPU Utilization. (2) Performance degradation due to VM migration from one Host to another. Same metrics has been considered by past researchers [6], but our result shows better improvement in maintaining SLA considering these two metrics. We considered SLAs are delivered when 100% requirement of an application has been served by a VM. However, when a Host experience CPU Utilization of 100%, the performance of Application degrade due to leaping the capacity.

\[
\text{SLA time per active Host} = \frac{1}{X} \sum_{p=1}^{X} H_{ap} \tag{4}
\]

\[
\text{Performance degradation due to migration} = \frac{1}{Y} \sum_{m=1}^{Y} V_{am} \tag{5}
\]

Where and are the number of Hosts and VMs respectively; is the total time when host has experienced 100% CPU utilization caused SLA violation; is the total time host was active; is the approximation of performance degradation due to migration of VM and we consider it 10%; is the total CPU capacity requested by the VM during its epoch.

3.2 Heuristic for dynamic resource allocation

The multidimensional bin packing problem can also be solved using heuristics. So, the problem of VM allocation can be divided in two ways. Those can be mentioned as- admission of new requirements for VM provisioning and enlisting the VMs in the host, and optimization of current VM allocation. The first part can be seen as a bin packing problem with variable bin size and prices, whereas bins represent the physical hosts, bin sizes are the available CPU capacity and bin process are the energy consumption. Although Heuristic solution will not guarantee an optimal solution, the required time to obtain a feasible solution is much shorter than LP. We solved the first problem by using Best Fit Decreasing Algorithm that is shown to use no more than \(11/9\) OPT + 1 bins [14]. However, for solving the second problem, we consider power consumption of Hosts and CPU utilization of Hosts. The pseudo-code of VM selection and VM placement is presented in the Algorithm. We calculate the other available host’s current CPU Utilization and increase of Power consumption if the selected VM had have migrated. By using \(\sqrt{x^2 + y^2}\), we get a point where it indicates the tradeoff point between CPU Utilization and incremental of Power Consumption of Hosts and find the suitable Host for VM. Otherwise it will find a Host which Utilization is higher calculating that the Host doesn’t get overloaded if the VM migrated to that Host.

```
Algorithm 1: VM Allocation Algorithm

Input: hostList vmList Output: Allocation of VM’s

foreach h in hostList do
    vmList ← h.getVmList ()
    vmList.sortDecreasingUtilization ()
    hUtil ← h.getUtil ()
    bestFitUtil ← MAX
    while hUtil > \(T_u\) do
        foreach vm in vmList do
            if vm.getUtil () > hUtil then
                t ← vm.getUtil () – hUtil + \(T_u\)
                r ← vm.getRam ()
                c ← sqrt (sqr (t) + sqr (r) )
                if c < bestFitUtil then
                    bestFitUtil ← c
                    bestFitVm ← vm
                end
            else
                if bestFitUtil = MAX then
                    bestFitVm ← vm
                    break
                end
        end
        hUtil ← hUtil - bestFitVm.getUtil ()
        migrationList.add(bestFitVm)
        vmList.remove(bestFitVm)
    end
foreach bestFitVm in vmList do
    minPower ← Max
    allocatedHost ← Null
    maxHost ← MIN
    mindiagonal ← MAX
    foreach host in hostList do
        if bestVmUtil() < THRESH_UP – hUtil() then
            powerdiff ← powerAfterAllocation – getPower (host)
            hUtil ← getUtilizationOfCpu (host)
            A← sqrt (sqr (powerdiff) + sqr (hUtil))
            if A < mindiagonal then
                allocatedHost ← A
            end
        else
            if hUtil > maxHost
                allocatedHost ← host
            end
```

3.3 QoE and SLA Violation

Addressing quality from the view points of the end user’s communication experience and their perceived QoE is relatively new approach to the Cloud Computing environment while so many efforts have been made to improve QoS of Networks to fulfill the targeted SLA. Although QoE is very subjective in nature, it is very important that a strategy is devised to measure it as realistically as possible (1). Based on the method we discussed earlier we make a subjective and quantitative formulation to get QoE, which is related to SLA violation. We build a relationship between QoE and SLA violation as our proposed approach that deals with dynamic VM allocation technique for proper resource management rather network parameter analysis. A typical user-related measure is the mean opinion score (MOS), which can be determined from subjective ratings by real users or predicted from objective measurements of properties of the delivered service [17], which is used as a part of our subjective analysis. So, the numerical formula to get QoE is as follows:

\[
QoE = e^{- (\frac{SLA_i}{\alpha})} \times \left( \sum_{i=1}^{n} W_i X_i^* - \alpha^+ + \alpha^- \right)
\]  (6)

where, \( C_{i,j,k} \in C \); \( C_i \neq C_j \neq C_k \) and \( W_i = 0 \sim 1 \)

Figure (2): Relationship between QoE and SLA Violation

In equation (6), \( SLA_i \) denotes the SLA violation rate during the workload execution; \( C_i \) represents the subscribed SLA class priority, if the subscribed service class is high, the constant will have high priority value. It means the QoE level of premium service subscriber’s will be lower comparing to other subscriber’s at same SLA violation rate; \( SLA_i \) symbolizes the subscribed SLA Class (Gold, Silver, Bronze). Different class will have different priority e.g. \( C_i, C_j, C_k \); we define Mean Opinion Score (MOS) by \( (\sum_{i=1}^{n} W_i X_i^* - \alpha^+ + \alpha^-) \), where \( W_i \) and \( X_i \) epitomizes \( i^{th} \) criterion and positive weight of \( i^{th} \) criterion respectively; \( \alpha^+ \) and \( \alpha^- \) are determined as overestimation and underestimation error.

Figure (2) depicts that, for the same SLA violation how QoE differs to different subscribed class. We set priority \( C_{i,j,k} = 1, 2, 3 \) for gold, silver and bronze class SLA having different subscriber value. QoE was measured at the scale of 3 having SLA Violation ranging from 20%-40%.

4. Performance Evaluation

4.1 Test bed Setup:

We evaluate our proposed algorithm in CloudSim toolkit. CloudSim is an extensible simulation toolkit that enables modeling and simulation of Cloud Computing systems and application provisioning system created by CLOUDS Lab, University of Melbourne [17]. We choose to do that as it is enormously difficult to conduct large-scale experiments in real infrastructure. 800 heterogeneous physical nodes is used to do the experiment. Three types of VM was used as High-CPU medium instance (2500 MIPS, 0.85 GB), Extra Large instance (2000 MIPS, 3.75 GB) Small instance (2000, 3.75 GB) and Micro Instance (1000 MIPS, 1.7). We have conducted experiment with real life data provided by the CoMon project, a monitoring infrastructure for PlanetLab. We have used CPU Utilization data of more than thousand VMs from different geographic Places. We have traced random 10 days’ workload data from March and April, 2011 and the average CPU load was below 70% and workload was assigned to VMs randomly. We compared with LR-MMT (Local Regression- Minimizing Migration Time) [6], BFD (Best Fit Decreasing), FFD (First Fit Decreasing) [7] to show the improvement of our result. LR-MMT method was so far the best approach in this area (1).

<table>
<thead>
<tr>
<th>Simulator</th>
<th>CloudSim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Host</td>
<td>800</td>
</tr>
<tr>
<td>Host features</td>
<td>Intel Xeon 3040, 2 cores × 1860 MHz, 4 GB, Intel Xeon 3075, 2 cores × 2660 MHz, 4 GB</td>
</tr>
</tbody>
</table>
4.2 Analysis

In order to compare the efficiency, we have evaluated SLA Violations with other researcher’s proposed algorithms. Our proposed approach outperforms the algorithm provided by Anton et al. [6] and solution proposed by [7]. From the experiment we calculated the amount of time an Active Host experienced 100% CPU Load that caused performance degradation for the application in its epoch and number of VM migration occurs due to VM consolidation. We haven’t done aggressive consolidation otherwise it has been resulted in other way, increasing the SLA violation. Due to dynamically selecting the Upper threshold and efficiently placing of VMs in the Hosts, lesser number of migrations and higher rate of maintaining SLA occur. Fig: (4.a) shows 8.5%, 52.63%, and 59% less SLA Violation comparing to LR-MMT algorithm, BFD and FFD algorithm while experimenting with 10 days’ workload with CloudSim. During the workload execution, LR-MMT, BFD and FFD algorithm suffers from 100% Host overloading by 2.3%, 5.6% and 4.3% than our proposed approach resulting more SLA Violation. Moreover, SLA Violation reflects on measuring QoE and our approach shows significant improvement. Fig: (4.c) indicates QoE enhancement of 9.3%, 23% and 25.9% comparing with LR-MMT, BFD and FFD respectively.

- Number of VMs: 1052, 898, 1061, 1516, 1078, 1463, 1358, 1233, 1054, 1033
- VMs Feature: Amazon EC2 instance type
- Workload Data: CPU Utilization From PlanetLab for 10 days
- Control Time: 5 minutes
5 Conclusions

In this paper we propose a demand prediction method based on the current usage of resource. In addition to that we also solved resource allocation problem heuristic solution. Performance comparison shows that our resource management/allocation approach performs very competitively providing better QoE. This work does not include the cost analysis model for different cloud providers, broker intelligence and how resource can be distributed by different types of Cloud Providers from different geographic sites. As for the future works, we would incorporate some of the above as a part of the future work.

6 Conclusions

This work was supported by a grant from the NIPA (National IT Industry Promotion Agency) in 2012. (Global IT Talents Program). Professor Eui-Num Huh is corresponding Author.

7 References


