SLA-Based Resource Allocation System in Cloud Computing

Ms.Ashwini Paygude, 1-4- UG students, 5- Professor Department of Information Technology Dnyanganaga College of Engineering and Research, Narhe, Pune, India

Ms.Sonali Deshmukh, 1-4- UG students, 5- Professor Department of Information Technology Dnyanganaga College of Engineering and Research, Narhe, Pune, India

Mr.Chetan Ghumare, 1-4- UG students, 5- Professor Department of Information Technology Dnyanganaga College of Engineering and Research, Narhe, Pune, India

Mr.Nilesh Korpad, 1-4- UG students, 5- Professor Department of Information Technology Dnyanganaga College of Engineering and Research, Narhe, Pune, India

Prof.Mr.P.M.Mane 1-4- UG students, 5- Professor Department of Information Technology Dnyanganaga College of Engineering and Research, Narhe, Pune, India

ABSTRACT--Cloud computing solves the problem of enterprises application distribution. To deliver hosted services to customers SaaS companies either maintain their own hardware or rent it from infrastructure providers. This requires extra cost. So to minimize a cost, this paper proposes resource allocation algorithm which will minimize cost and SLA violations. Algorithms also ensure the mapping of customer requirements into infrastructure level parameters. We also consider customers quality of service parameters, dynamically changing customer demands and handling of the customer needs.The resource allocation policies assure the full utilization of each VM capacity. It also assures the customer satisfaction level and minimization of penalty caused by violation of SLA.

Keywords : Cloud Computing, Software as a Service, Service Level Agreement, Scheduling.

I. INTRODUCTION

In traditional model, when customer needs any software, he has to purchase the license copy of that software. In addition to that customer needs technical expertise who will handle the issues regarding the installation and maintenance of the software. If there are any updates of software then customer requires to pay additional cost. With existence of cloud computing model, customer do not purchase the software. Instead they rent the usage of software from the third party. They use the software for the time they required and pay only for the time they used it i.e. they use software on pay-as-you-use basis. This minimizes the cost of customer and also enables the customer to focus on his core work. This saves the customer from the license fee for new versions and complexity of maintaining softwares. SaaS providers calculate their profit from the margin between infrastructure cost and revenue generated from the customer. Therefore SaaS providers need a solution that will minimize the infrastructure cost without affecting the customer satisfaction level.

In SaaS model of cloud computing, customer request for the software services provided by SaaS provider. There are three layer namely application layer, platform layer and infrastructure layer. Application layer contains the software services that are hosted on cloud for customers which will be used on pay-as-you-use basis. Platform layer includes mapping and scheduling policies. This layer includes mapping policies to map customer request into infrastructure level parameter and assigning VM's for each customer request. Infrastructure layer includes the VM related task such as adding and removal.

In current scenario, SaaS provider like Compiere ERP provides an individual VM to serve request of one customer. This minimizes SLA violation and saves them from penalty. But the main disadvantage of this is that it wastes capacity of individual VM i.e. capacity of individual VM can not be fully utilized. This also increases the cost of infrastructure which minimizes the profit.

Therefore to achieve the SaaS providers objective of maximizing profit and minimizing the SLA violations to provide customer satisfaction some resource allocation strategies are proposed. In this strategies various factors are taken into consideration such as dynamically changing customer needs and QoS parameters.

II. RELATED WORK

Reig G. et al [1] in his research provides services by minimizing the resource for providing client request and executes them before its deadline. But if the system available resources fails to provide services to clients request before its deadline then this system allows scheduling policies to discards the service of request.

Popovici et al. [2] mainly considered QoS parameters onthe resource provider's side such as price and offered load,but did not focus on the user side.

Lee et al. [3] investigated the profit driven service requestscheduling for workflow. In contrast, our work focuses on a) SLA driven QoS parameters on both user and provider sides,and b) solves the challenge of dynamic changing customerrequests to gain profit and improve reputation.

For enterprise applications, Song et al. [4] presented the genetic algorithms in virtualized environments. However, the genetic algorithms generally require a long execution time. The long execution time increases the probability of SLA violation in the Cloud computing environments, where customers need to be served immediately.

Fu Y. et al [5] proposed an SLA-based dynamic scheduling algorithm (Squeeze) of distributed resources for streaming.

So the key points in this paper will be as follows.

1. It includes the mapping and scheduling policies which assures customer satisfaction and minimizes VM to be required which in turn increases the profit of SaaS provider.
2. It considers QoS parameters such as response time.
3. It considers the dynamically changing customer demands.

III. SYSTEM ARCHITETURE

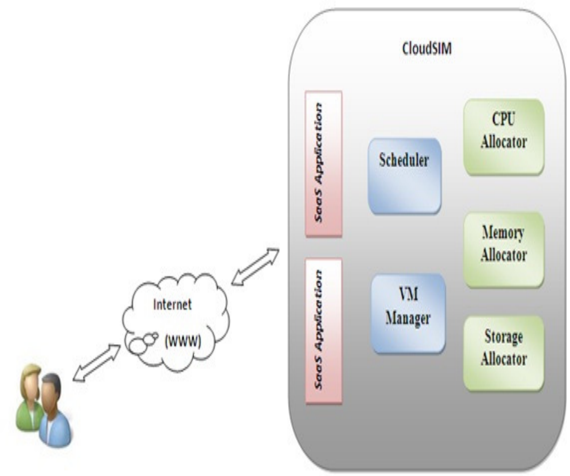

Figure 3.1 System Architecture

User :

User is customer who have signed SLA for using softwares or SaaS application that are hosted by SaaS provider.

CloudSim:

CloudSim goal is to provide a generalized and extensible simulation framework that enables modeling, simulation of cloud computing infrastructure and application services so that user can focus on design issues.

VM Manager manages the initation and removal of VM's .Scheduler schedules the requests.

IV. SYSTEM IMPLEMENTATION

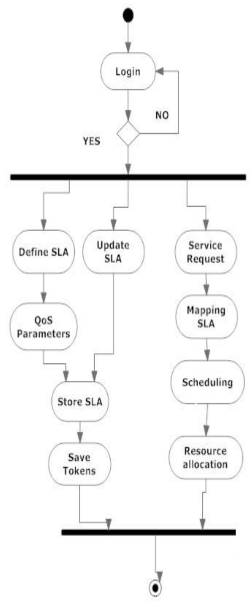

Figure 4.1 System Flow

We consider customer request for software services from a SaaS provider by agreeing to the pre-defined SLA clauses and submitting their QoS parameters. Customers can dynamically change their requirements and usage of the hosted software services.The parameters defined in SLA will be as follows.

Request Type (*reqType*): It defines the customer request type, which is 'fist time rent' or 'upgrade service'.

Product Type (*proType*): The software product offered to customers. For example, *SaaS X* offers Standard, Professional, and Enterprise product.

Account Type (*accType*): It constrains the maximum number of accounts a customer can create. For example, *SaaS X* offers three types of account: Group, Team, and Department, which allows each customer to create up to *m*, *2m* and *5m* number of accounts respectively.

Contract Length (*conLen*): How long the software service is legally available for a customer to use (minimum is one month).

Number of Accounts (*accNum*): The actual number of accounts that a customer wants to create.

Number of Records (*recNum*): The maximum number of records a customer is able to create for each account.

Response Time (*respTime*)

Table 1.Mapping of customer QoS requirements to resources.

VM type	Product type	Account Type	Max Account #	Min Account #
Small	standard	group	m	1
medium	standard, professional	team	2m	m+1
large	Standard, professional,enterprise	department	5m	2m+1

ALOGORITHM

The main objective of our work is to maximize the profitfor a SaaS provider by minimizing the cost of VMs usingeffective platform layer resource allocation strategies.

This algorithm reduces the number of violations by usinga new VM for each company to guarantee the response time.However, it is costly because a large number of VMs areinitiated.

Maximizing the profit by reusing VMs (*ProfMaxVmMinAvaiSpace*).

Algorithm
1. Pseudo-code for *ProfMaxVmMinAvaiSpace*
Inputrequest (*c*) with QoSparameters ,VM*i*
OutputBoolean
Functions:
request (*c*) with QoS parameters , VM*i*
Boolean
First Time Rent (*c*){
1 If (there is initiated VM***i***with type *l* matches to the VM type
requested by *c*) {
2 If (VM*i*deployed the same product type as *c* required) {
3 For each initiated VM***i***with type *l* (VM*il*){
4 If (VM***i***has enough space to place *c*)**{**
5 put VM*i*into *vmList*
6 }
7 }
8 Sort(*vmList***)** according to the available space
9 Schedule to process *c* on ***VMmin***, which has minimum
available space i.e. using the best fit manner.

10 }
11 Else {
12 Initiate new VM with type *l* and deploy the product type as request *c* required
13 }
14 }
15 Else While (*l+j*<=*L*) loop {
16 If (there is initiated VM with next type *l+j,* where type *l+j*matches to the VM type required by request *c*) {
17 Repeat from Step 2 to 13
18 j++
19 }
20 }
21}
Upgrade(*c*) {
1 If (upgrade type is 'add account'**) {**
2 get Id *i*and type *l* of VM, which processed the previous request from same company as *c*
3 If (VM*i*has enough space to place *c*){
4 Schedule to process *c* on VM*i*.
5 }
6 Else {
7 Repeat step 1 to 21 of **First Time Rent(*c*)**
8 Transfer data from old VM to new VM
9 Release space in old VM
10 }
11 }
12 If (upgrade type is **'upgrade service'**){
13 Repeat step 7 to 9 of **Upgrade(c)**
14 }
15}

V. CONCLUSION

The proposed system will minimize the number of VM's which will lead to maximizing the profit of SaaS provider. It also considers the Customer satisfaction level. It will give the expected level of customer satisfaction and hence will minimize the penalty which is result of SLA violation. It will handle the dynamic change in customers request and will work properly for this.

VI. REFERENCES

[1] G Reig, J. Alonso,and J. Guitart, "Deadline Contrained Prediction of Job Resource Requirments to Manage High-Level SLAs for SaaS Cloud Providers". Tech. Rep. UPC-DAC-RR, Dept. d'Arquitectura de Computadors, University Polit'ecnica de Catalunya, Barcelona, Spain.

[2] I. Popovici, and J. Wiles, "Proitable services in an uncertain world". In Proceeding of the18th Conference on Supercomputing (SC 2005), Seattle, WA.

[3] Y.C. Lee, C. Wang, A.Y. Zomaya and B.B. Zhou, "Profit-driven Service Request Scheduling in Clouds". In Proceedings of the International Symposium on Cluster and Grid Computing, (CCGrid 2010), Melbourne, Australia.

[4] Y. Song, Y. Li, H. Wang, Y. Zhang, B. Feng, H. Zang, Y. Sun, "A Service-Oriented Priority-Based Resource Scheduling Scheme for Virtualized Utility Computing", High Performance Computing-HiPC 2008.

[5] Y.Fu, A. Vahdat, "SLA Based Distributed Resource Allocation for Streaming Hosting Systems", http://issg.cs.duke.edu