JCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-3, Issue-5 E-ISSN: 2347-2693

# Obviate SLA violation for Resource Allocation based on SaaS in Cloud Environment

Faraz Khatri[1], Faraz Shaikh[2], Deepak Yadav[3*], Pramod Murkute[4]

[1,2,3*,4]*Al-Ameen College of Engineering, Pune, Maharashtra, India*

**www.ijcseonline.org**

Received: Apr/26/2015 Revised: May/06//2015 Accepted: May/22/2015 Published: May/30/ 2015

***Abstract***— Cloud Computing is an emerging way of storing data, but still it on the verge of improving with every development. Taken in an account for performing more profit, as providing services to various users are difficult because it should in a manner though all can benefit in their work. Allocation is tedious task due to: 1) Diversity in resource allocation. 2) Hard to map request 3) frequent change in customer. Cloud level of agreement done throughout the Service Provided, SLA provide a standard way of controlling the mechanism. This paper deal with an allocation of resource considering a token like price, response time, space and contract length. As all the type of user can benefit the more of the system resource.

***Keywords***—SLA (Service Level Agreement); Resource Allocation; Cloud Computing;

## I. INTRODUCTION

E-Commerce growth rate is increasing with high rate, through Cloud Computing considered as a solution for many enterprises distribution application which can solve the traditional sales model. diaspora from tradition model to cloud lead to an on-going revenue. as main focus is on the the allocation of the process of the user request. this lead to the system to perform more efficient and reliable and robust through which the SAAS provider can satisfy all Quality of Service parameter[2].Also cloud computing presents many advantages, such as flexible scalability, pay-as-you-use, high power conservation and economies of scale. Deploying traditional business services on Virtual Machine (VM) instances rented from one or more provider. Here an SLA came into existence, A SLA is a contract between a service provider and a user, which is a collection of service level requirements that simply specify to promise service performance and the corresponding profits, conventionally speaking, SLA include such properties as response time, user budget and reliability, elixirs for performance failures.

A three-tier cloud service provision structure has been formed involving three typical parties: Infrastructure service providers, cloud service providers and end users [1]. Cloud provides these services in two paradigm IAAS (Infrastructure as a Services) and PAAS (Platform as a Services), which help of Software services in application layer. An overview of this paradigm is as follow by their definition.

Infrastructure as a service: IaaS is a type of cloud computing in which a third-party provider hosts virtualized computing resources over the Internet. Resource is available at all instances which can be scaled down as per the need.

Software as a Service: SaaS is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet.

Platform as a service: PaaS is a cloud model in which providers deliver apps over the Internet and host users' hardware and software on their infrastructure.

The cloud service provisioning process can be simply described as follows: above all, the end user sends there service request to the cloud service provider [1], then cloud service provider accepts the user request and applies to the underlying cloud infrastructure. Hosting centers host services of the clients on their behalf and guarantee quality of service [3]. The cloud infrastructure responds to resource lease request, and then allocates VM instances to the corresponding cloud service provider for processing the cloud user request and then cloud service provider charges end user for processing service request and pays the cloud infrastructure vendor for renting VM instances to deploy service capacity [1]. Perfect things are hard to gain due to reasons like: resource discrepancy and require of more energy rate. Quality of service as defined by service level agreements to reduce energy misuse and to increase profit ideal allocate resources to meet client SLAs. Ideal allocation is a non-trivial task due to two factors: Hosting centers have wide resource variety where energy usage of a client task varies depending on the allocated resources. Second, due to lack of energy rate also energy cost for a task varies based on server utilization as which lead to meet the higher energy rate to achieved ideal and good allocation of resource and the service provider also get most of the benefit of it service provided.

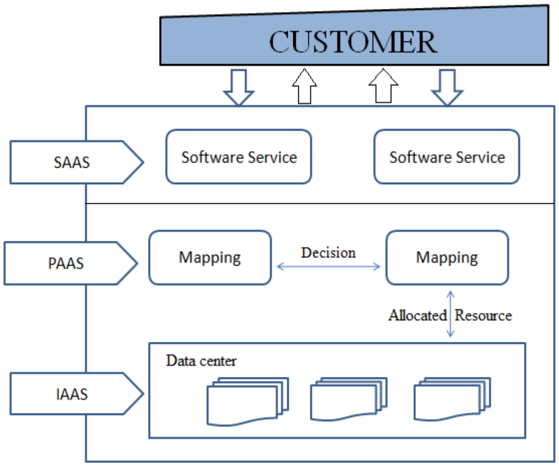

Figure.1 System model for SAAS

## II. RELATED WORK

Previously work on resource allocation is most done by keeping in mind to increase the profit and to minimize the cost of the service provider. Dynamically allocation of resource, change of request, customer request, cost of mapping, scheduling policies and response time are the main policy for building a good SLA quality parameter.

The key point of this paper as follows:-

- To improve quality of parameter.
- Mapping customer request.
- Ability to change /manage customer request.
- Manage to reduce penalties.
- Good response time.
- Build an Mechanism to reduce infrastructure cost and maximize profit.

This paper focus on service arrival time and rate of request by customer, also initial time and penalty by SaaS provider.

## III. SYSTEM MODEL

Considering an customer request for a software service from SaaS provider by approving the defined SLA provision with all the quality of Service parameter with Dynamically change in requirement and usage of resource. As main purpose of SaaS provider is to list itinerary work as per SLA guideline at registration due to which the system is responsible for scheduling which job to be executed and at which particular place, so system should be manageable and able to execute properly throughout the performance and should not degrade with extra job or nodes. In an dynamic situation the whole system should be configured and allow multiple scheduling queries that should satisfy the whole quality of service parameter, VM can route as per the SLA dynamic load so it can be managed as per the given job. As minimizing the occurrence of SLA violation while resource allocation, so that customer request can be sort to grow to a less costing.

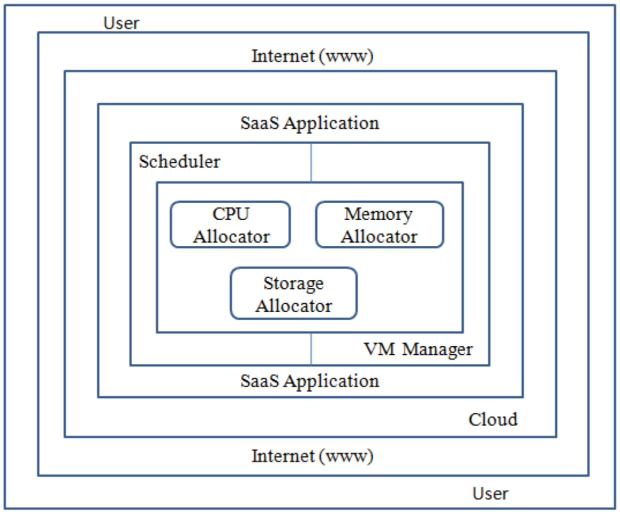

Figure.2 Cloud Deployment Model

As customer request is for an enterprises software service from a SaaS provider where the customer has agreed with the defines SLA policy which classify the whole quality of service parameter also customer can dynamically change its requirement as per their use and upgrade the service anytime they needed. The scheduling mechanism is design so it will minimize the Infrastructure cost and SLA violation between both by determining which VM to be initiated in the particular time by keeping in policy clauses on basics of price, initial time and time require to transfer the data. As response time play a crucial matter so that the particular job can be finished faster and correctly so to maximize the profit to customer services and SaaS provider without violating any clauses in the system. The two algorithm play an important part through which the system work. (i)SmxA: the Vm which contain the maximum space (ii)SmiA: the Vm which contain minimum space related to account.

SLA properties can be defined as: (i) Request type (ii) Product type (iii) Account Type (iv) Contract Length (v) Response time.

*1)* **Request Type (slaRequest):** It defines the type of reuest ask by customer as it is new service or any upgrade service

*2)* **Product Type (slaProduct):** which type of software product offered to customer, like enterprises or professional.

*3)* **Account Type (slaAccount):** maximum number of account to a customer can create. Like group, team etc.

*4)* **Contract Length (slaLength)**: actual number at account that customer have to create.

*5)* **Account Count (slaAccNO):** actual number of account that customer want to create.

## IV. Strategies Model

### *A.* Scheduling Strategies

Figure.2 shows the deployment model of the whole process where the whole cloud is consider into account where SaaS, PaaS, IaaS also an customer, in which any user can access the SaaS application through any web portal, as the request is made the CPU allocator came into account which lead to memory allocation and storage allocation is consider. If the requests meets the valid pass through then Scheduler allocate the particular VM is allocated in VM manager in PaaS layer. Now the whole balance the load of system and manage the allocation process.

### *B. Mathematical Model*

Set Theory
Set Theory Analysis
Let 'A' be the "SLA-based Resource Allocation in cloud"
A= {…………….}
Set A is divided into 6 modules
A= {A1, A2, A3, A4, A5, A6}
A1= GUI Handlers (GH)
A2= Configuration Manager (CM)
A3= VM Manager (VMM)
A4= SLA Manager (SLAM)
A5= Policy Manager (PM)
A6= Database Manager (DM)

Identify the inputs.
Inputs = {I1, I2, I3, ……..In}
I1= SLA
I2= Account addition
I3= Update SLA

Identify the output as O.
Outputs = {O1, O2, O3, ……..On}
O1= VM Allocation
O2= Resource Allocation

**Problem Description:** Let S be a system which do SLA-based Resource Allocation in cloud; such that S = {A1, A2, A3, A4, A5, A6} where A1 represents GUI Handlers (GH) Module; A2 represents Configuration Manager (CM) Module; A3 represents VM Manager (VMM) Module; A4 represents SLA Manager (SLAM) Module; A5 represents Policy Manager (PM) Module; A6 represents Database Manager (DM) Module.

***UML Design Observations***

S holds list of modules in the system

Activities:

*Activity I: SaaS Provider SLA Creation*

Let A1 be a set of SaaS provider parameters for SLA creation.
A1= {user_id, sla_id, sla_values}
Where,
user_id: user's id
sla_id: defined SLA id
sla_values: SLA values

| Condition/Parameter | Operation/Function |
|---|---|
| If no change in SLA data | Discard value |
| Else. | f1:Proceed() |

**Table 1. SaaS Provider SLA Creation**

UML Design observations:
If user's SLA info is not updated or changes then please discard the value Else read the SLA value any send data to database.

*Activity II: Allow number of accounts as per SLA defined*

Lets A2 be a set of accounts policy details parameters:
A2= {user_id, sla_id, sla_type, account_count}
Where,
user_id: user's ID
sla_id: SLA ID
sla_type: Type of SLA like basic, enterprise, etc…
account_count: No of account allowed in SLA type

**Table 2. Allow number of accounts as per SLA defined**

UML Design observations:
Search in the user's SLA policy that the user's can create account or not. If the user's SLA policy allow then add account. Else don't add account and show error messages.

*Activity III: Allocation of VM as per SLA*

Let A3 be a set of user's VM allocation parameters:
A3= {user_id, sla_id, sla_type, vm}
Where,

| Condition/Parameters | Operation/Function |
|---|---|
| account_count | f1:searchSLAPolicy(); |
| If (user is allowed to add account) | f2:checkSLAPolicy() ; |
| Add Account | f3:CreateAccount(); |
| Else, discard account add info | |

user_id: user's ID
sla_id: SLA ID
sla_type: Type of SLA like basic, enterprise, etc…
vm = Virtual machine to be added on host

| Condition/Parameters | Operation/Function |
|---|---|
| Vm_policy | f1:searchVMPolicy(); |
| If (user is allowed to add VM on host) | f2:checkVMPolicy(); |
| Allocated and add new VM as per available space on host | f3:AllocateVM(); |
| Else, discard users VM request | F4: discard() |

**Table 3. Allocation of VM as per SLA**

UML Design Observations:
Search in the user's SLA policy that the user's VM should be added. If the user's policy allow then add VM. Else don't add VM and throw error.

*Activty I: Show Result graphs*
Let S4 be a set of parameters required for graph generation.
S4= {user_id, sla_id, reports}
Where,
user_id: user's ID
sla_id: SLA ID
reports= Reports to be Displayed

UML Design Observations:
Here dynamically graphs will be shown to user.

## V. CONCLUSION

There is a revolution in cloud computing which has been split in research for managing the quality of service for better efficiency of resource scheduling. This paper has focused on high priority on resource utilization and minimizing the cost and maximizes the profit of software as a service provider. As dynamically changes of customer request can be addressed concurrently, lead to efficiently focus on mapping of the request to infrastructure level. In future work, we will like add more parameter and services in scheduling the application which will maximize the profit of service provider and efficiently allocation of resource without SLA violation.

## Acknowledgment

We think anonymous reviewers and Staff of Al-Ameen College of Engineering, Pune who has helped us to improve the quality of paper.

## REFERENCES

[1] 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.

[2] Linlin Wu, Saurabh Kumar Garg and Rajkumar Buyya, "SLA-based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments" in Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing.

[3] Kimish Patel and Murli Annavaram, "NFRA:Generalized Network Flow-Based Resource Allocation for Hosting Centers" in IEEE TRANSACTIONS ON COMPUTERS, VOL 62, NO. 9, SEPTEMBER 2013.

[4] Vincent C. Emeakaroha, Ivona Brandic, Michael Maurer and Ivan Berskovic, "SLA-Aware Application Deployment and Resource Allocation in Clouds", in proceedings of 35th IEEE Annual Computer Software and Applications Conference Workshops, 2011.

[5] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, Software: Practice and Experience (SPE), Volume 41, Number 1, Pages: 23-50, ISSN: 0038-0644, Wiley Press, New York, USA, January, 2011.

[6] C. Vecchiola, X.C. Chu, M. Mattess, and R. Buyya, "Aneka Integration of Private and Public Clouds", Cloud Computing Principles and Paradigms, Willy, USA, 2011

## AUTHORS PROFILE

Faraz Khatri
Department of Information Technolgy,
Al-Ameen College of Engineering,
University of Pune.

Faraz Shaikh
Department of Information Technolgy,
Al-Ameen College of Engineering,
University of Pune.

Deepak Yadav
Department of Information Technolgy,
Al-Ameen College of Engineering,
University of Pune.