IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Issue-1, Jan 2019 E-ISSN: 2347-2693

Template-Based Efficient Resource Provisioning and Utilization in Cloud Data-Center

Seema Chowhan[1], Ajay Kumar[2], Shailiaja Shirwaikar[3]

[1]Department of Computer Science, Baburaoji Gholap College, India
[2]Jayawant Institute of computer Application, Pune, India
[3]Department of Computer Science, Savitribai Phule Pune University, Pune, India

**Corresponding Author: ssc_chowh@yahoo.com, Tel.: 020-27280204*

Available online at: www.ijcseonline.org

Accepted: 24/Jan/2019, Published: 31/Jan/2019

Abstract— Cloud computing, with an ever-growing interest, with the promise of revolving computing as a utility after water, electricity, gas and telephony is currently at a stage, where many enterprises are considering adapting to this technology. Resource provisioning policies allow efficient sharing of resources available in a data center and these policies help to evaluate and enhance the cloud performance. Resource provisioning that maintains quality of service with optimum resource utilization is a challenge. It is a multidimensional problem that can have issue based solution in the form of a set of services that help allocation and negotiation of service level agreements. A cloud simulator environment is used and experiments are performed by varying different parameters of Virtual Machines (VM) and the tasks running on VM, to get optimal values for designing templates. The proposed template based resource provisioning (TBRP) method overcomes under-provisioning and over-provisioning of resources for agreed parameters specified by SLA.

Keywords— Service Level Agreement, Quality of Service, Virtual Machines, Resource Provisioning.

I. Introduction

Cloud is a pool of huge and heterogeneous infrastructure of resources, to be shared over the Internet by a large set of users with dynamically changing requirements. Cloud computing is abstraction of web based resources and services for high performance computing in businesses having dynamic requirements of resources with reliability, cost-effectiveness and availability. It is evolution of variety of technologies that bundles together to provide IT infrastructure as per organization's needs [1], [2]. Cloud provider has to create an illusion of availability of unlimited computing resources to the end users on limited hardware and unpredicted request loads. The challenges for cloud computing provider is to allocate resources as per Service Level Agreements (SLAs) and performance of cloud system should be stable in any dynamic changes of workload as per SLA specified without effecting quality of service (QoS). By Byun et al [3], SLAs specify the resources and quality levels required for the execution of job in order to minimize the cost from user perspective and to maximize the resource utilization from provider's perspective. In such systems Quality of service parameters are availability, reliability, response time and throughput in contractual documents agreed between provider and customer called SLA [3] [4], [5], [6], [7].

In case of load variation with constant resource there is a need to have improved methods and techniques to overcome the problems of Over-Provisioning and Under-Provisioning.

- Over-Provisioning: SLA is satisfied and conflicts are avoided, but large set of resources are left idle leading to unnecessary costs.
- Under-Provisioning: Provisions efficiently utilize the existing resources but are insufficient to guarantee the agreed Quality of Service (QoS) leading to frequent breaches in SLA.

In case of load variation there is a need to understand the variation in the response time. The customer and Data-center will understand response time variation and required resources on the variation of workload time to time before designing SLA.

Lack of methodologies for VM provisioning raises a risk that all VMs deployed on a single host may not get the sufficient amount of processor share that is crucial for fulfilling the agreed SLAs. The proposed Template-Based Resource Provisioning (TBRP) method will give the insight of response time and optimum utilization of idle capacity on variation of workload. Rest of the paper is organized as follows, Section I contains the introduction of Template Based Resource provisioning and utilization, Section II contain the background and related work of SLA based resource provisioning, virtualization, Section III contain proposed TBRP method procedures, Section IV Pre-Set up and various experiments for parameter extraction, section V explain the testing of TBRP method, Section VI concludes research work with future directions.

II. BACKGROUD AND RELATED WORK

The recent development in cloud computing and pay per use model enables procurement of large bundle of computational and storage resources on request basis [8]. Quality of service parameters agreed between provider and customer are documented in a contractual form called service level agreement (SLA) which ensures delivery of QoS parameters, such as availability, reliability, response time and throughput to the users as per signed agreement [4],[5], [6], [7].

II.1 Virtualization

One of the important strengths of cloud is its infrastructure management, empowered by progress of virtualization technology, to better utilize the underlying resources through automatic provisioning and balancing of workload. Computing environments can be dynamically created, scaled up, scaled down or moved as user workload fluctuates [9],[10]. Virtualization is software that separates physical infrastructure into logical partitions. It operates and controls hardware that is physically distributed by sharing computing resources from collections of servers and dynamically assigning virtual resources to applications on-demand [1],[11]. A virtual machine is nothing but a virtual server that combines a set of physical resources like CPU core, RAM Storage and bandwidth to create various dedicated resources. In cloud computing, users access services as per their requirements irrespective of where they are hosted. Cloud provider provides different deployment models based on service types. The four deployment models for using resources of cloud include the public cloud, private cloud, hybrid cloud, and community cloud [12]. Virtualization improves agility, elasticity, minimizes cost and thus enhances business value for provider. Cloud provides server, storage and network virtualizations [13], [14].

II.2 Resource Provisioning and Utilization

Load Balancing can be carried out both at resource provisioning level which is heavily dependent on the Service Level Agreements (SLA) and also at the resource utilization level. Cloud provider needs to optimally provision the resources to enlarge the market share and to maintain customer satisfaction level by avoiding penalty payments. Major challenges in the resource allocation and resource management in cloud computing environment includes resource modeling, resource offering and treatment, resource discovery and monitoring and resource selection [15].

Several researches have addressed these requirements by providing SLA based resource management mechanism. SLA management system provides, benchmarking of cloud performance and present a way to measure and incorporate performance information into SLAs [16]. A model called LoM2HiS is proposed by Emeakaroha et al for efficient resource management to improve availability of resources. The framework facilitates autonomic SLA management and enforcement which detects future SLA violation threats and can notify to act so as to avert the threats [17]. A Meta scheduler that optimizes resource utilization in terms of number of jobs meeting their deadlines (QoS) has been proposed by Jeyarani et al [18]. A federated cloud mechanism and a technique that describes broker architecture for effective management of users to be linked to the best available cloud service providers, with interoperability through brokers is investigated by Rajarajeswari et al [19]. Adaptive QoS-aware virtual machine provisioning mechanism has been developed by Feng et al [20] that ensures efficient utilization of the system resources by linking QoS to low-level infrastructure resource and serving all the tasks within the requirements described in SLA. A task-oriented multi-objective scheduling method based on ant colony optimization has been proposed to optimize the resources in a hybrid cloud environment by providing deadline and cost as constraints [21]. In addition to this Zuo et al also proposed a multi-objective optimization scheduling method using an improved ant colony algorithm to achieve optimization of both performance and cost according to the makespan and the user's budget costs as constraints to optimization problem [22]. Tambe et al propose an approach for efficient resource sharing in heterogeneous network by assigning the priorities to task which results in lesser completion time of task [23].

A typical workload in cloud systems will be mixture of heterogeneous applications. The number of clients to large enterprise software systems/websites such as (e.g. Amazon, Facebook) is highly varying on number of aspects such as time of day, day of week and other seasonal factors. Capacity planning of such application is highly challenging, resulting into under provisioning or over provisioning.

Optimal resource provisioning problem is challenging due to diversity in client and QoS requirements of different applications are different [24]. Transactional applications required response time and throughput, Non interactive batch processing applications required turnaround time (Completion time) and throughput whereas real time applications required response time as QoS requirement. Sebagenzi et al [25] proposed integrated approach for load balancing and energy efficiency of Data center with scheduling reservations of virtual machine (VM) for CPU-intensive applications.

III. PROPOSED TEMPLATE-BASED RESOURCE PROVISIONING (TBRP) METHOD

The proposed TBRP method is to minimize cost, provisioning of resources as per SLA, monitoring of resource utilization and guarantee the response time to assist design of SLA. In this procedures of resource provisioning and utilization strategy system is designed such that VMs are utilized sparingly at low workload and judiciously at high work load to maintain SLA by maintaining Quality of Service (QoS) levels.

In this TBRP method the entire resource is divided into small, medium and large VM types and the combination of these VM types are designated as Templates for provisioning resources which periodically maps idle capacity into a set of VM templates for cost constraint and efficient resources utilization.

The method have three different MIPs rating (processing capacity) that are small-VM, medium-VM and large-VM from which VM templates are so designed that it will meet users SLA requirements and helps in managing overprovisioning and under-provisioning. In case of customer need data-center will provide available templates to satisfy customer requirements or data-center will create custom made templates to meet customer requirement. The method also utilizes full capacity of Data Center to avoid over-provisioning and under-provisioning while giving attractive pricing for users. The Cloud service provider can also maximize profit without affecting the customer satisfaction.

The templates can be defined as per the capacity of Data-Center. The small-VM, medium-VM and large-VM can be different for different capacity of Data-Centers. In this paper the concept is tested experimentally with the MIPs rating of 100, 150 and 200 as small-VM, medium-VM and large-VM using CloudSim simulator and the results are analyzed. The proposed TBRP procedure is prescribed in the Figure-1.

III.1 Procedure of Proposed TBRP Method

This method start with defining the VM templates cost followed by provisioning resources for such templates and monitoring resources utilization keeping in view of QoS.

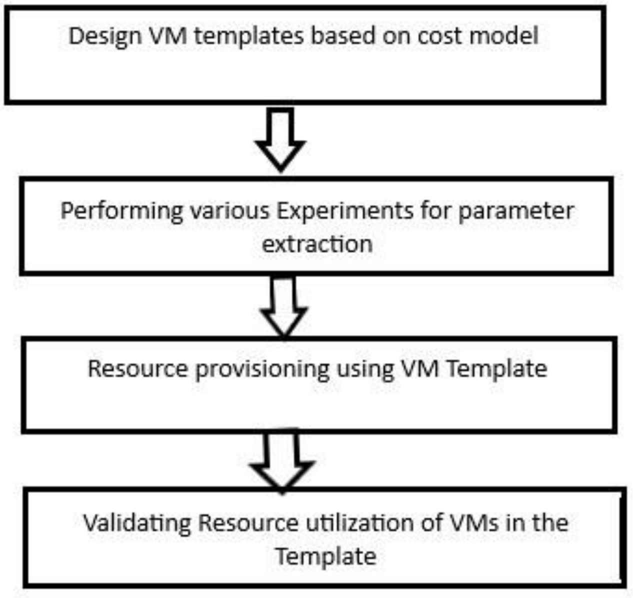
Design VM templates based on cost model
Performing various Experiments for parameter extraction
Resource provisioning using VM Template
Validating Resource utilization of VMs in the Template

Figure-1: Proposed TBRP procedure

III.1.1 Design VM Templates Based On Cost Model

Step-1: Defining VMs cost as per SLA requirements- That is task Completion time. The cost of VM is the cost per unit of time define as

$$CostVM^{i} = a(MIPs)+b(bandwidth)+c(RAM)+ d(storage)+e(No-of-proces\sin gelement) \quad (1)$$

Step-2: Defining Total Cost of VM template by collecting the simulation data over a time. Let α= Number of small-VM, β = Number of medium-VM and γ= Number of large-VM .The number of VMs of each type (α, β, γ) are also to be fixed once for all so that the combined cost is effectively smaller than the expected Return on Investment (ROI) for the resources.

$$\mathrm{TotalCost} = \alpha \mathrm{CostVM}^{\mathrm{small}} + \beta CostVM^{medium} + \gamma CostVM^{large} < ExpectedROI \quad (2)$$

α = 1, 2…….p, β =1, 2……..q, γ =1, 2……..r

The total of p+q+r should be less than and equal to 'Total idle capacity of Data-Center' for the tasks assigned in that Data Center.

III.1.2 Performing Various Experiments for Parameter Extraction

Step-1: Different experiments are performed for resource provisioning and utilization.

Step-2: The result of variation in various parameters like number of virtual machines (VM), attributes of VMs such as MIPs rating, number of tasks and attributes of tasks such as length, has been studied.

Step-3: The completion time is then used to infer optimal parameter values for designing and utilizing templates.

III.1.3 Resource Provisioning Using VM Template

Step-1: Defining VM template to meet QoS parameters values as per below specifications in Table-1 to guaranteed SLA.

Table-1: QoS based VM Templates configuration

VM template
Configuration-VM(s): Specifying the number small-VM, medium-VM and large-VM Type
QoS Constraints setting as per SLA clauses: Response Time, Completion Time, Maximum Workload, any other parameters as per requirements
Calculation of Pricing
Setting of the Penalty

Step-2: Defining completion time as per SLA clauses as in table-2 for different workloads, so that it should not lead to SLA violations. The various constants used in the SLA clauses need to be learned from simulation experiments executed over a period for the perfection of SLA design.

Table-2: Completion Time SLA clauses

Completion Time SLA clauses
0 ≤ workload < LowWorkrload ⟹ Completion Time <minCTvalue LowWorkload≤ workload <MedWorkload ⟹ Completion Time <medCTvalue MedWorkload≤ workload <MaxWorkload ⟹ Completion Time <maxCTvalue

Step-3: The pricing strategy for VM template apart from the cost of its components, takes into consideration several other factors such as type of service, time of the day of the service, customer standing etc.

$CostVM^{Template} = n_1CostVM^{small} + n_2CostVM^{medium} + n_3cost\ Vm^{large} \pm Charge^{ServiceType} \pm Charge^{ServiceTime} \pm Charge^{CustomerType} \pm$ ……...

A resource provisioning strategy can provision the resources as per customer requirement as and when workload increases, avoid SLA violations and minimize utilization cost.

III.1.4 Validating Resource Utilization of VMs in the Template

Step-1: Table-3 is Resource utilization procedure at a given instant that will checked as per Predicted load in the form of number(s) and size(s) of the tasks. The next steps is to start and gradually using VMs by distributing the load proportionately. The proportionate load distribution is the ratio of predicted workload divided by Total capacity.

Table-3: Resource utilization strategy

```
Resource utilization strategy
Input : Predicted workload and the template
configuration
QoS parameters in SLA/ VM template:
MinCTvalue...LowWorkload...
Algorithm:
If predicted workload  is less than LowWorkload
       Start and use a single small VM
      if completion time exceeds minCTvalue then
         completion Time SLA violation
else
 if predicted load is less than MedWorkload
      Start and use  both small  and medium VM by
      distributing the load proportionately
     if completion time exceeds medCTvalue then
         completion Time SLA violation
else
 if predicted load is less than MaxWorkload
     Start and use all the three VMs by
     distributing the load proportionately
     if completion time exceeds maxCTvalue then
        completion Time SLA violation
else
Maxworkload SLA violation
```

Step-2: The above resource utilization strategy gets appropriately modified when the template configuration is much more complex where in there can be zero or more VMs of each type.

For low workload (0 <=workload<Low-workload)
    If there is small-VMs then
        All small VMs will start.
    Else if there are no small-VMs then
        All medium size VMs will start.
    Else if there are no small-VMs and medium-VMs
        All large size VMs will start.
The load is then proportionately divided among these VMs.

Step-3: At medium work load all small VMS and medium size VMs will start but if there are no small or medium size VMs then all large size VMs will start.

Step-4: At high workload (workload >= Medium-workload) all VMs of each type will start and the load will be divided proportionately amongst them.

The Resource utilization strategy thus uses provisioned VMs sparingly at the same time takes care that there are no SLA violations.

IV. PRE SET-UP AND PARAMETER EXTRACTION

In CloudSim environment, Datacenter consists of fixed or varied configuration of hosts (servers). The hosts, VMs and cloudlets in Data-Center are characterized with attributes and configurations as indicated in Table-4. The software as a service requires parameters like MIPs, bandwidth and processing elements whereas IaaS requires RAM, storage and number of processing elements. As we are working with SaaS during experiment, we assume that cloud resource users only request for virtual nodes only. The experiment will simulate how virtual nodes which are on a same data center deal with multiple tasks. Each task or application represents a user's request; which can dynamically increase as per the requirement. Effects of varying VM and cloudlet parameters are checked for task completion time.

Table-4: Configurations of cloud elements

Data center	Host	Virtual Machine	Cloudlets

time_zone = 10.0
cost of processing=3.0
costPerMem = 0.05
costPerStorage = 0.001
costPerBw = 0.0
storageList
	Ram: 20000 MB
	Storage: 2000000 MB
	No of Processing Entities: 6
	Bandwidth 30000
		MIPS rating
		Image Size on Disk
		RAM
		Bandwidth
		Number of Required Pes
			Length (in terms of instructions)
			Input File Size
			Output File Size

IV. 2 Modeling of VMs Characteristics

For simulation experiments, completion time was chosen as the QoS constraint. The simulation experiment is conducted to understand and test VM parameters Image size, RAM, Bandwidth and Pes-number and their impact on completion time by varying different VM parameters and load. The computing load is in the form of number of cloudlets of varying cloudlet length.

Further in this experiment 500 cloudlets having 40000bytes of cloudlet length are executed on varying MIPs values from 50-MIPs to 500-MIPs by change value of 50-MIPs on different size of VM(s) to observed completion time of VMs and the results are tabulated in Table-5 for the better selection of MIPs values for small-VM, medium-VM and large-VM.

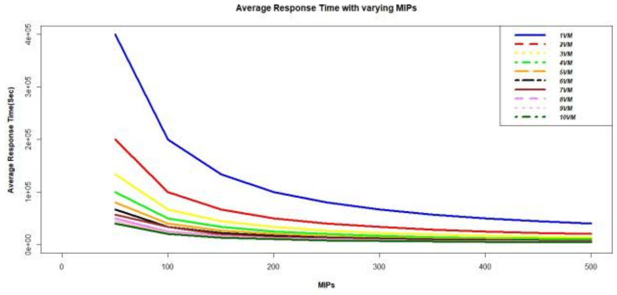

Figure-2: Average completion time with varying MIPs

Observations*:* From the experiments results researcher observed that Image size, RAM, Bandwidth and Pes-number characteristics of VM have not impacted on completion time of task. Only MIPs values shows comparable difference in completion time.

Data center can select MIPs value as per completion time in designing templates as tabulated in Table-5 .The results in Figure-2 indicates increase in MIPs value leads to decrease in completion time and better performance. Since the tasks are equally distributed among number of allotted VMs, the completion time is same for a single VM of MIPS value 500 and 10 VMS of MIPS value 50. As template constituents, three VM types of varying capability are to be chosen and as MIPS characterize the performance in terms of completion time, three different MIPs values are to be chosen for three VM types. Three values of 100,150 and 200 are selected as MIPs rating for small, medium and large VM in design of VM templates.

The cost of VM type depends on the MIPS rating and the parameter a in equation-1 is significant and it can be computed a ≈ Expected ROI / Total capacity in MIPS

IV. 3 Experiments for Testing of QoS Constraints for Varying Load

The computing load can be varied by varying the number of cloudlets to be executed or by varying the cloudlet length. In the first experiment the cloudlet length is varied by keeping the number of cloudlets to be executed as fixed. Thus 500 cloudlet were executed on single VM of small, medium and large type. Alternatively the computing load was varied by increasing the number of cloudlets of fixed cloudlet length.

Table-6: Completion time with fixed cloudlets and Varying Cloudlet length

Cloudlet-length	Single-LVM(Sec)	Single-MVM(Sec)	Single-SVM(Sec)
10000	24998.33	33331.67	49996.67
20000	49998.33	66665	99995.83
30000	75000	100000	150000
40000	99998.33	133331.7	199996.7
50000	124998.3	166665	249995.8
60000	150000	200000	300000

Table-5: Varying MIPs with varying number of VMs with completion time

VM capacity (MIPs)	**Completion Time (Sec) for 1-VM to 10-VM(s)**									
	1-(VM)	**2-(VM(s))**	**3-(VM(s))**	**4-(VM(s))**	**5-(VM(s))**	**6-(VM(s))**	**7-(VM(s))**	**8-(VM(s))**	**9-(VM(s))**	**10-(VM(s))**
50	**400000.1**	**200000.1**	**133331.3**	**100000.1**	**80000.1**	**66667.27**	**57144.3**	**50002.12**	**44447.09**	**40000.1**
100	**200000.1**	**100000.1**	**66665.69**	**50000.1**	**40000.**	**33333.7**	**33333.7**	**25001.09**	**22223.62**	**20000.1**
150	**133333.4**	**66666.77**	**44443.85**	**33333.43**	**26666.77**	**22222.48**	**19048.18**	**16667.45**	**14815.76**	**13333.43**
200	**100000.1**	**50000.1**	**33332.88**	**25000.1**	**20000.1**	**16666.9**	**14286.15**	**12500.61**	**11111.88**	**10000.1**
250	**80000.1**	**40000.1**	**26666.37**	**20000.1**	**16000.1**	**13333.55**	**11428.96**	**10000.51**	**8889.53**	**8000.1**
300	**66666.77**	**33333.43**	**22221.97**	**16666.77**	**13333.43**	**11111.31**	**9524.17**	**8333.78**	**7407.96**	**6666.77**
350	**57142.96**	**28571.53**	**19047.44**	**14285.81**	**11428.67**	**9524**	**8163.6**	**7143.25**	**6349.71**	**5714.39**
400	**50000.1**	**25000.1**	**16666.55**	**12500.1**	**10000.1**	**8333.52**	**7143.13**	**6250.36**	**5556.02**	**5000.1**
450	**44444.54**	**22222.32**	**14814.68**	**11111.21**	**8888.99**	**7407.58**	**6349.46**	**5555.88**	**4938.7**	**4444.54**
500	**40000.1**	**20000.1**	**13333.23**	**10000.1**	**8000.1**	**6666.84**	**5714.53**	**5000.31**	**4444.84**	**4000.1**

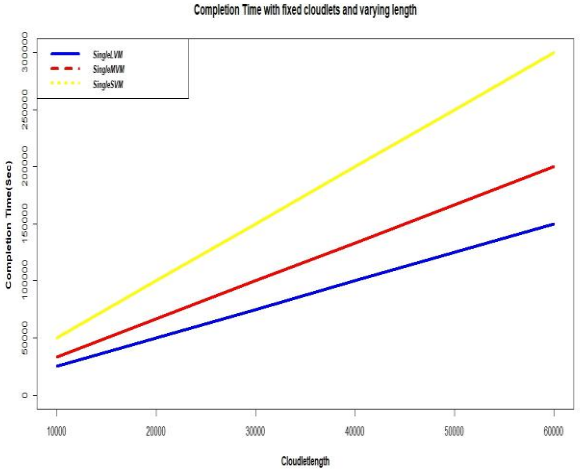

Figure-3: Completion time with fixed cloudlets and Varying Cloudlet length

Table-7: Completion time with fixed Cloudlet length and Varying Cloudlets

Cloudlets	Single-LVM (Sec)	Single-MVM (Sec)	Single-SVM (Sec)
32	1599.89	2133.17	3199.73
64	3199.76	4266.29	6399.41
125	6249.58	8332.92	12499.17
250	12499.17	16665.83	24998.33
500	24998.33	33331.67	49996.67
1000	49996.67	66663.33	99993.33

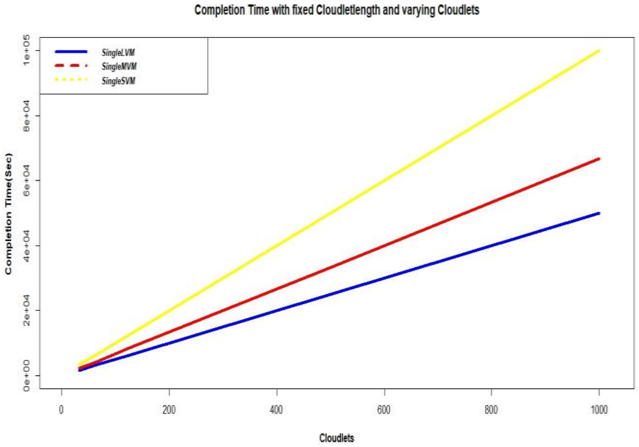

Figure-4: Completion time with fixed Cloudlet length and Varying Cloudlets

Observations: The results in Table-6 and Figure-3 indicate that the completion time linearly depends on cloudlet length by executing fixed cloudlets. The results in Table-7 and Figure-4 indicate that the completion time linearly depends on cloudlets by keeping cloudlet length fixed .Thus computing load can be defined in terms of cloudlet length multiplied by number of cloudlets. The completion time is better on large VMs as compared to Medium and Small VMs.

IV. 4 Extracting and Fixing SLA Parameters for VM Templates

In designing templates, apart from the configuration one need to define the SLA clauses in the QoS constraints. The parameters used in these can be derived by performing simulation experiments. Consider a VM template of simple configuration comprising of one small, one medium and one large VM. The computing load depends on the application and varies at different point in time. In general four scenarios are considered. The application is having low workload that is when there are few tasks of small size. The workloads is medium when the both task size and number is same or one of them is small and the other is not very large. The load reaches the pick when either large size tasks get executed or tasks increase in number. The load is extreme when both size of the task and number increase beyond limit. In the simulation experiment, workload for these four situations are executed on VM template of single small, medium, large VM and also on VM of combined MIPs size. In case of template the load is distributed proportionately. The experiment results presented in Figure-5 show that the time taken on the template is very close to the time taken on the VM of same size.

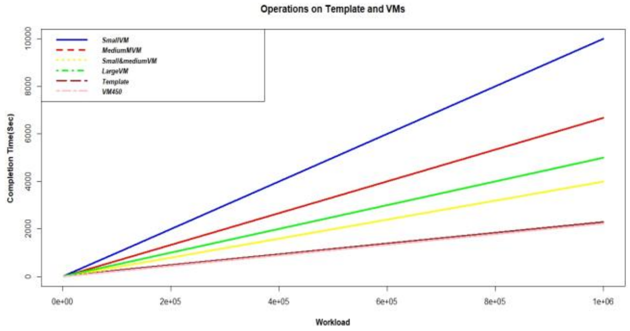

Figure-5: operations on VM (small VM-SVM, medium VM-MVM and large VM-LVM) and Template

The four workload situations are shown in the Table-8 where the workload is the product of cloudlet length and number of cloudlets. The completion time on the different VMs is used to extract SLA parameters that will avoid SLA violations when using resource utilization strategy.

Table-8: workload scenarios

User Id	No of Cloud-Lets	Cloudlet Length	Completion Time (Sec)					
			Single SVM	Single MVM	Single SVM+ MVM	Single LVM	Template	VM 450 MIPs
1	10	100	10	6.67	3.97	5	2.49	2.22
2	100	100	100	66.67	39.73	50	22.81	22.22
3	10	10000	1000	666.67	399.97	500	249.99	222.22
4	100	10000	10000	6666.67	3999.73	5000	2298.88	2222.22

V. TESTING OF TBRP METHOD

After setting up Data-Center and parameters extraction the method is tested on completion time for single VM templet with VM-size of 450(MIPs). The method is also tested for comparative analysis on completion time and VM-Utilization having same capacity and varying capacity VM-Template MIPs.

V.1 Performance Comparison of Single VM (450 Mips) With VM-Templates-SML

The experiment is conducted to compare completion time of single VM of 450 MIPs with VM-Templates-SML (having one small-VM, one medium-VM and one large-VM) at low work load ranging from 200 to 1Kbytes, medium work load ranging from 2 to 10 Kbytes and large work load from 20Kbytes to 100 Kbytes. The results on completion time and resource utilization are tabulated in Table-10 and presented in Figure-6.

Mathematical model of Utilization (VMs in VM Template):

Utilization (Small-VM) = (Number of small-VMs)* (MIPs Capacity of small-VMs) / Total MIPs Capacity of VM Template (3)

Utilization (Medium -VM) = (Number of medium-VMs)*(MIPs Capacity of medium-VMs) / Total MIPs Capacity of VM Template (4)

Utilization (Large -VM) = (Number of large-VMs)*(MIPs Capacity of large-VMs) / Total MIPs Capacity of VM Template (5)

The various constants used in the SLA clauses learned from simulation experiments are presented in Table-9.

Table-9: SLA clause constants with values

Workload Type	Workload Value	Completion Time	CT value(Sec)
Low workload	1000	minCTvalue	10
Medium workload	10000	medCTvalue	40
Large workload	100000	maxCTvalue	250

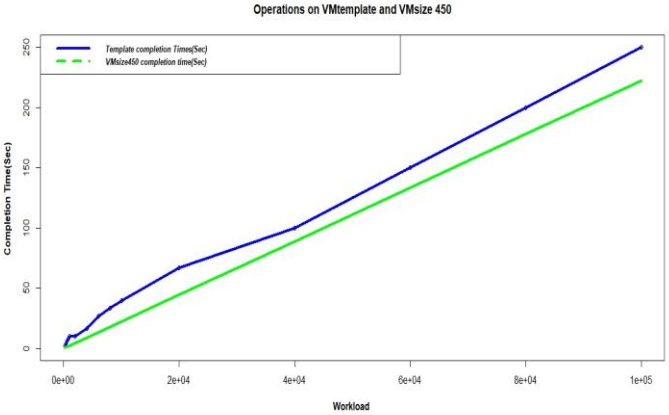

Figure-6: Completion time on VM template and VM of size 450 MIPs

The VM-Templates utilization is calculated and presented in Figure-7 to understand utilization of VMs in templates.

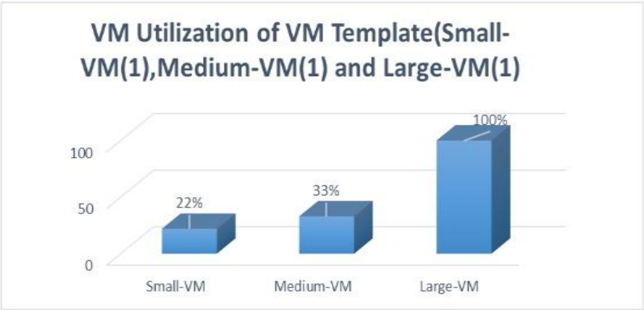

Figure-7: VM Utilization of VM Template (Small-VM (1), Medium-VM (1) and Large-VM (1)

Table-10: Completion time on VM template and VM-size of 450 MIPs

Cloudlets (Number of task)	Cloudlet length (byte)	Workload (bytes)	Completion Time (Sec)	
			VM-Templates-SML	VM-Size 450(MIPS)
Low-workload				
2	100	200	2	0.44
4	100	400	3.99	0.89
6	100	600	5.96	1.32
8	100	800	7.95	1.77
10	100	1000	10	2.22
Medium-workload				
2	1000	2000	9.99	4.44
4	1000	4000	16.66	8.89
6	1000	6000	26.65	13.32
8	1000	8000	33.3	17.77
10	1000	10000	39.73	22.22
Large-workload				
2	10000	20000	66.66	44.44
4	10000	40000	99.96	88.88
6	10000	60000	149.99	133.32
8	10000	80000	199.99	177.77
10	10000	100000	249.99	222.22

Observations: After simulation experiment the completion time from Table-10 are maximum 10 seconds for low workload, 40 seconds for medium workload and maximum 250 seconds for large workload. From Table-11 the experiment results shows that the completion time in all these three cases of workload are below the set completion time. It is also seen that the execution of task on VM-template-SML completion time is little more than the completion time on VM (450 MIPs) but VM utilization is better in case of template method from figure-6. Since the cost of VM is proportional to its MIPs values. From figure-6 one can observe that at low workload 22% and at medium workload 33% of the provisioned capacity of Template is utilized while at high workload 100% provisioned capacity is utilized. Thus resource utilization is minimized at means that cost is saved and SLA is not violated.

V.2 Comparative Analysis of Varying VM-Templates with Different MIPs Capacity

The different simulation experiments are performed for comparative analysis on completion time and VM-Utilization having same capacity of VM-Template MIPs or varying capacity of VM-Template MIPs on different combination of VM-Templates.

V.2.1 VM Templates of 600-Mips Capacity Completion Time on Different Workload

The simulation experiment is conducted to understand the performance and utilization of VM templates with different configuration but all having 600-MIPs capacity. The three different VM-Templates has been taken in that the Templates are of only small or medium or large types and three VM-Templates are mixture of small, medium and large VMs. The T-SML is different combinations of Templates. The T600 means only having small VMs, T040 means only having medium VMs, T003 means only having large VMs, T320 means having small and medium VMs, T202 means having small and large VMs and T121 means having small, medium and large VMs. The completion time of these VM-Templates are tabulated in Table-11 and presented in Figure-8 for observations, analysis and comparisons on completion time.

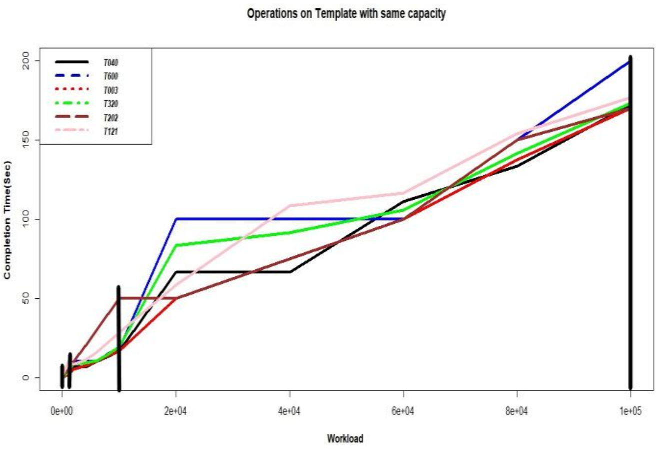

Figure-8: VM-Templates (T040, T600, T003, T320, T202 and T121) completion time performance on Varying loads

Observations: From figure-8 it is observed that though all the templates have the same total capacity, they vary considerably in their performance at different workloads. The templates T040, T003 and T320 perform uniformly at different workloads while T600 performs poorly at high workloads while T003 gives best performance at high workloads.

For clear understanding of VM Templates performance on completion time for low workload, medium workload and large workload the Figure-8 is spread and presented in Figure-8(a) for Low workload, Figure-8(b) for Medium workload and Figure-8(c) for High workload.

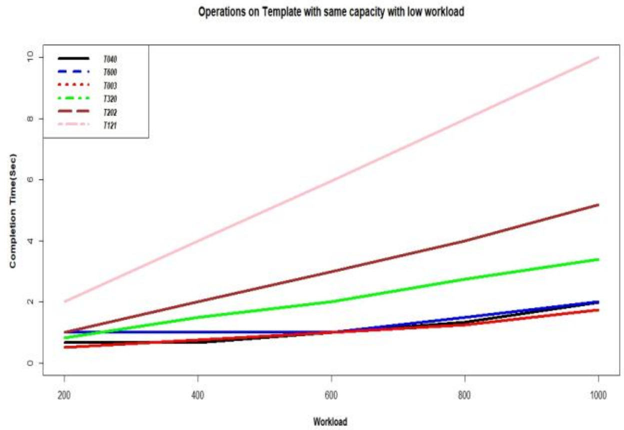

Figure-8(a): VM-Templates (T040, T600, T003, T320, T202 and T121) completion time performance on low workload

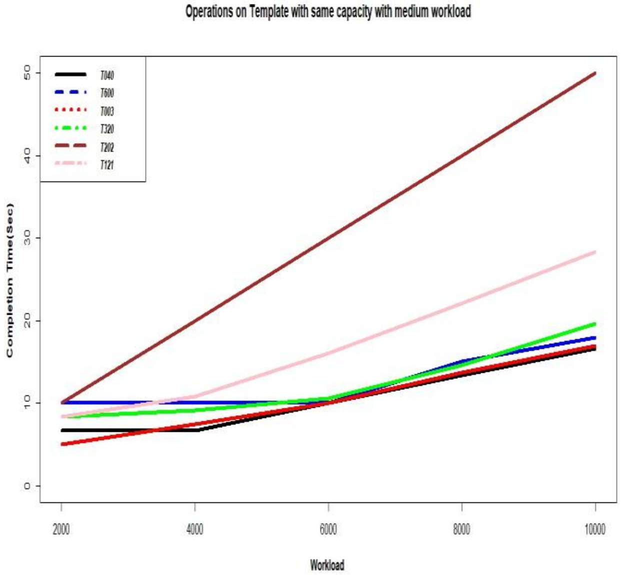

Figure-8 (b): VM-Templates (T040, T600, T003, T320, T202 and T121) completion time performance on medium workload

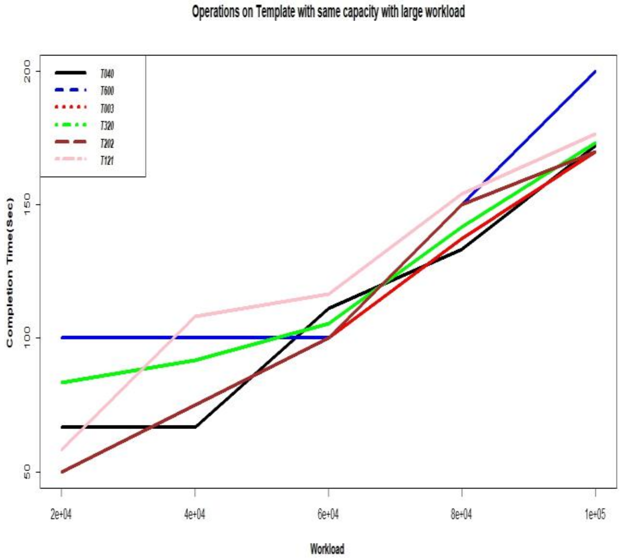

Figure-8 (c): VM-Templates (T040, T600, T003, T320, T202 and T121) completion time performance on large workload

Table-11 Completion time on varying VM templates with same capacity

Cloudlets (Number of task)	Cloudlet length (byte)	Workload (bytes)	600 MIPs Capacity completion time (sec)					
			VM-Templates size of only small or medium or large			VM-Templates that is mix of small, medium and large		
			T040	**T600**	**T003**	**T320**	**T202**	**T121**
Low workload								
2	100	200	0.66	1	0.5	0.83	1	2
4	100	400	0.66	1	0.75	1.5	2	3.99
6	100	600	1	1	1	2	2.98	5.96
8	100	800	1.33	1.5	1.24	2.74	3.99	7.96
10	100	1000	**1.99**	**2**	**1.74**	**3.38**	**5.17**	**10**
Medium workload								
2	1000	2000	6.66	10	5.0	8.33	10	8.33
4	1000	4000	6.67	10	7.5	9.16	20	10.83
6	1000	6000	10	10	10	10.55	29.97	16.1
8	1000	8000	13.33	15	13.74	14.57	39.99	22.07
10	1000	10000	**16.66**	**18**	**16.99**	**19.6**	**49.99**	**28.32**
Large workload								
2	10000	20000	66.66	100	50	83.33	50	58.33
4	10000	40000	66.66	100	75	91.66	75	108.32
6	10000	60000	111.11	100	100	105.55	100	116.66
8	10000	80000	133.33	150	137.49	141.66	150	154.16
10	10000	100000	**172.33**	**199.99**	**169.99**	**173.32**	**169.99**	**176.65**

Observations: Figure-8(a) indicates that for low workload templates T040, T003 and 006 performs uniformly at different workloads while T121 performs poorly at low workload. Figure-8(b) indicates that for medium workload the templates T040, T003, T320 and 006 performs uniformly with slight difference in completion time values whereas T202 performs poorly at medium workload. Figure-8(c) it is seen that for large workload that the templates T040, T003 and T320 perform uniformly at different workloads while T600 performs poorly at high workloads while T003 gives best performance at high workloads.

V.2.1.1 Completion Time Range of SLA Clause

The six Templates completion time range on various load are extracted from Table-12 and presented in Table-13.

Table-12: SLA clause VM template variables with values

Workload Types (W) bytes	Completion time of Templates (sec)					
	T040	T600	T003	T320	T202	T121
L=1000	1.99	2	1.74	3.38	5.17	10
M=10000	16.66	18	16.99	19.6	49.99	28.32
H=100000	172.33	199.99	169.99	173.32	169.99	176.65

The completion time range of SLA clauses on above defined VM-Templates are presented in Table-13 are extracted from Table-12 for different workload scenarios of SLA that can be given to different user's applications at cloud Data-Center better utilization and to meet QoS.

V.2.1.2 Utilization of VM Templates on 600 Mips Capacity

The VM-Templates (T040, T600, T003, T320, T202 and T121) utilization is calculated as per previously defined equation-3, equation-4 and equation-5 and presented in Table-14 and Figure-9 to understand utilization of VMs in templates.

Table-14: VM utilization of VM template (Small-VM, Medium-VM and large-VM) with same capacity

Template	Low workload (%)	Medium Workload (%)	High workload %)	Average utilization (%)
T040	100	100	100	100
T600	100	100	100	100
T003	100	100	100	100
T320	50	100	100	83.33
T202	33.33	33.33	100	55.55
T121	16.66	66.66	100	61.10

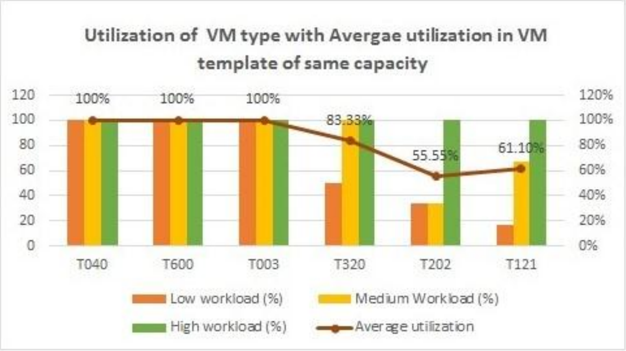

Figure-9: VM utilization for same capacity template

Table-13: Completion time range of SLA clause for varying VM template with same capacity

Template	Completion Time range of SLA clauses
T040	$(0 \le W < L \Longrightarrow$ Completion Time $< 2)$AND$(L \le W < M \Longrightarrow$ Completion Time$<17)$AND $(M \le W < H \Longrightarrow$ Completion Time $<173)$
T600	$(0 \le W < L \Longrightarrow$ Completion Time $< 2)$AND$(L \le W < M \Longrightarrow$ Completion Time$<18)$AND $(M \le W < H \Longrightarrow$ Completion Time $<200)$
T003	$(0 \le W < L \Longrightarrow$ Completion Time $< 2)$AND$(L \le W < M \Longrightarrow$ Completion Time$<17)$AND $(M \le W < H \Longrightarrow$ Completion Time $<170)$
T320	$(0 \le W < L \Longrightarrow$ Completion Time $< 4)$AND$(L \le W < M \Longrightarrow$ Completion Time$<20)$AND $(M \le W < H \Longrightarrow$ Completion □Time $<175)$
T202	$(0 \le W < L \Longrightarrow$ Completion Time $< 6)$AND$(L \le W < M \Longrightarrow$ Completion Time$<50)$AND$(M \le W < H$ $\Longrightarrow$ Completion Time $<170)$
T121	$(0 \le W < L \Longrightarrow$ Completion Time $< 10)$AND$(L \le W < M \Longrightarrow$ Completion Time$<29)$AND$(M \le W < H$ $\Longrightarrow$ Completion Time $<177)$

Observations: Provisioned capacity is better utilized at different workload situations, when the template configuration contains different VM types as presented in Figure-9. Templates T121and T202 performance may not be the best but the resource utilization is better results in saving cost.

V.2.2Varying VM Template with Varying Capacity of MIPs Values

The simulation experiment is conducted to understand the performance and utilization of VM templates with template configuration of mixed VM types and having different total MIPs capacity of 550, 600, 650, 750 and 900. The five different VM-Templates T211, T121, T112, T212 and T222 has been taken such that each one having small-VMs, medium-VMs and large-VMs. The completion time of these VM-Templates are tabulated in Table-15 for different workloads.

Table-15: Completion time on varying VM templates with different capacity

Cloudlets (Number of task)	Cloudlet length (byte)	Workload (bytes)	Completion Time (Seconds)				
			550 MIPs	600 MIPs	650 MIPs	750 MIPs	900 MIPs
			T211	T121	T112	T212	T222
Low workload							
2	100	200	1	2	2	1	1
4	100	400	2	3.99	3.99	2	2
6	100	600	2.98	5.96	5.96	2.98	2.98
8	100	800	3.99	7.96	7.96	3.99	3.99
10	**100**	**1000**	**5**	**10**	**10**	**5**	**5**
Medium workload							
2	1000	2000	10	8.33	8.33	8.33	8.33
4	1000	4000	11.66	10.83	16.66	11.66	8.33
6	1000	6000	18.32	16.1	24.98	18.32	12.21
8	1000	8000	23.32	22.07	39.98	23.32	17.49
10	**1000**	**10000**	**31.97**	**28.32**	**41.65**	**31.97**	**23.32**
Large workload							
2	10000	20000	66.66	58.33	50	58.33	50
4	10000	40000	79.16	74.99	66.66	79.16	58.33
6	10000	60000	119.43	116.66	94.44	86.1	72.22
8	10000	80000	158.32	154.16	133.32	108.33	91.66
10	**10000**	**100000**	**189.99**	**176.65**	**161.66**	**136.66**	**123.32**

The VM-Templates (T211, T121, T112, T212 and T222) completion time performance on varying loads are presented in Figure-10

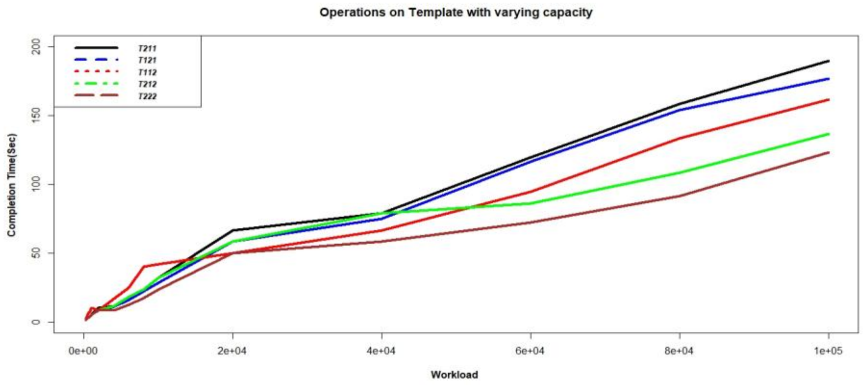

Figure-10 Operations on VM template with varying MIPs capacity

Observations: From simulation experiments as presented in Figure-10 it is observed that though all the templates have the different total MIPs capacity, they vary considerably in their performance at different workloads. Templates T121, T112 and T212 performs equally well slight difference in completion time values whereas T222 gives best performance at all workload scenarios. Template 211 performs poorly at high workload.

The VM-Templates (T211, T121, T112, T212 and T222) utilization is calculated as per previously defined equation-3, Equation-4 and equation-5 and presented in Table-16 and Figure-11 to understand utilization of VMs in templates.

Table-16: VM utilization of VM template (Small-VM, Medium-VM and large-VM) with varying capacity

Template	Template Capacity (MIPs)	Low workload (%)	Medium Workload (%)	High workload (%)	Average utilization (%)
T211	550	36.36	63.63	100	66.66333
T121	600	16.66	66.66	100	61.10667
T112	650	15.38	38.46	100	51.28
T212	750	26.66	46.66	100	57.77333
T222	900	22.22	55.55	100	59.25667

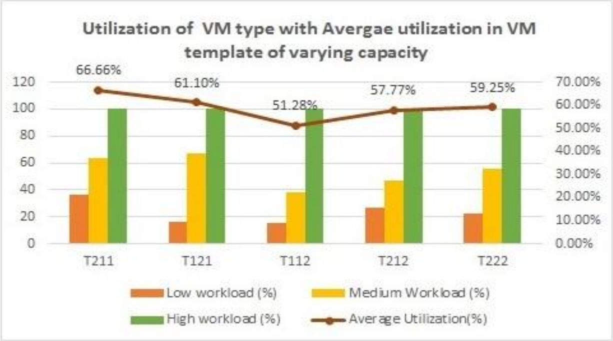

Figure-11: VM utilization on varying capacity templates

Observations: It is observed from Figure-11 that the resource utilization is better using resource utilization strategy when the template configuration is of mixed type with varying capacity in terms of MIPS value. Each template will have its own SLA constraints and it does not only depend on the MIPs capacity. Provisioned capacity is better utilized at different workload situations, as template configuration is of mixed type.

VI. CONCLUSION

The proposed template based resource provisioning and utilization method and procedure overcomes the problem of over-provision and under-provision of resources at Data-Center without effecting QoS and SLA on any workload. The method is driven by completion time as one of QoS parameter for different workload scenarios. The resource provisioning strategy using VM templates with predefined SLA clauses provides several options for negotiation, prevents SLA violations and allows better utilization. The TBRP method provides response time, resource utilization on load variation for designed SLA. This method is useful to extract SLA parameters by simulation before agree to SLA documentations. However, actual usage data can be used to design and offer better template for subsequent period on variation of load and response time. The TBRP method can be tested for comparative analysis on different capacity of Data centers.

REFERENCES

[1] Sosinsky, B. (2010). Cloud computing bible (Vol. 762). John Wiley & Sons.

[2] Shawish, A., &Salama, M. (2014). Cloud computing: paradigms and technologies. In Inter-cooperative collective intelligence: Techniques and applications (pp. 39-67). Springer Berlin Heidelberg.

[3] Byun, E. K., Kee, Y. S., Kim, J. S., &Maeng, S. (2011). Cost optimized provisioning of elastic resources for application workflows. Future Generation Computer Systems, 27(8), 1011-1026.

[4] Bianco, P., Lewis, G. A., & Merson, P. (2008). Service level agreements in service-oriented architecture environments (No. CMU/SEI-2008-TN-021). Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.

[5] John, M., Gurpreet, S., Steven, W., Venticinque, S., Massimiliano, R., David, H., & Ryan, K. (2012). Practical Guide to Cloud Service Level Agreements.

[6] Wu, L., &Buyya, R. (2012). Service level agreement (sla) in utility computing systems. IGI Global, 15.

[7] Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). NIST cloud computing reference architecture. NIST special publication, 500(2011), 292.

[8] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., &Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6), 599-616.

[9] Kremer, J. (2010). Cloud Computing and Virtualization. White paper on virtualization.

[10] Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., &Zagorodnov, D. (2009, May). The eucalyptus open-source cloud-computing system. In Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (pp. 124-131). IEEE Computer Society.

[11] Malhotra, L., Agarwal, D., &Jaiswal, A. (2014). Virtualization in cloud computing. J Inform Tech SoftwEng, 4(2), 136.

[12] Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications, 1(1), 7-18.

[13] Leavitt, N. (2009). Is cloud computing really ready for prime time. Growth, 27(5), 15-20.

[14] Rimal, B. P., Choi, E., &Lumb, I. (2009). A Taxonomy and Survey of Cloud Computing Systems. NCM, 9, 44-51.

[15] Endo, P. T., de Almeida Palhares, A. V., Pereira, N. N., Goncalves, G. E., Sadok, D., Kelner, J.,&Mangs, J. E. (2011). Resource allocation for distributed cloud: concepts and research challenges. IEEE network, 25(4).

[16] Gillam, L., Li, B., &O'Loughlin, J. (2014). Benchmarking cloud performance for service level agreement parameters. International Journal of Cloud Computing 2, 3(1), 3-23

[17] Emeakaroha, V. C., Brandic, I., Maurer, M., &Dustdar, S. (2010, June). Low level metrics to high level SLAs-LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments. In High Performance Computing and Simulation (HPCS), 2010 International Conference on (pp. 48-54). IEEE.

[18] Jeyarani, R., &Nagaveni, N. (2012). A Heuristic Meta Scheduler for Optimal Resource Utilization and Improved QoS in Cloud Computing Environment. International Journal of Cloud Applications and Computing (IJCAC), 2(1), 41-52.

[19] Rajarajeswari, C. S., &Aramudhan, M. (2014). Ranking Model for SLA Resource Provisioning Management. International Journal of Cloud Applications and Computing (IJCAC), 4(3), 68-80

[20] Feng, Y., Zhijian, W., & Qian, H. (2016). A novel QoS-aware mechanism for provisioning of virtual machine resource in cloud. *Journal of Algorithms & Computational Technology*, *10*(3), 169-175.

[21] Zuo, L., Shu, L. E. I., Dong, S., Zhu, C., & Hara, T. (2015). A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. *IEEE Access*, *3*, 2687-2699.

[22] Zuo, L., Shu, L., Dong, S., Chen, Y., & Yan, L. (2017). A multi-objective hybrid cloud resource scheduling method based on deadline and cost constraints. IEEE Access, 5, 22067-22080

[23] G.U.Tambe1, P.R. Bhaladhare2 "Efficient Resource Sharing in Heterogeneous Environments" International Journal of Scientific Research in Network Security and Communication, Vol.5, Issue.3, pp.123-127, **2017**.

[24] Garg, S. K., Gopalaiyengar, S. K., &Buyya, R. (2011, October). SLA-based resource provisioning for heterogeneous workloads in a virtualized cloud datacenter. In International conference on Algorithms and architectures for parallel processing (pp. 371-384). Springer, Berlin, Heidelberg.

[25] Sebagenzi Jason, Suchithra. R, "Scheduling Reservations of Virtual Machines in Cloud Data Center for Energy Optimization", International Journal of Computer Engineering, Vol.6, Issue.6, pp.16-26, **2018**.

Authors Profile

Ms. Seema Chowhan is working as a faculty and head in subject of computer science in Baburaoji Gholap College Pune, India affiliated to Savitribai Phule Pune University, Pune. She has 18+ years of experience in teaching UG and PG courses. She has completed M.Phil (CS).Her research interests include Cloud Computing and Networking.

Dr. Ajay Kumar experience covers more than 26 years of teaching and 6 years of Industrial experience as IT Technical Director and Senior Software project manager. He has an outstanding academic career completed B.Sc. App. Sc. (Electrical) in 1988, M.Sc. App.Sc. (Computer Science-Engineering and Technology) in1992 and PhD in1995. Presently, working as Director at JSPMs Jayawant Technical Campus, Pune (Affiliated to Pune University). His research areas are Computer Networks, Wireless and Mobile Computing, Cloud computing, Information and Network Security. There are 74 publications at National and International Journals and Conferences and also worked as expert, appointed by C-DAC to find Patent-ability of Patent Applications in ICT area. Six commercial projects are completed by him for various companies/ Institutions. He holds variety of imperative position like Examiner, Member of Board of Studies for Computer and IT, Expert at UGC.

Dr. Shailaja Shirwaikar has a Ph. D. in Mathematics of Mumbai University, India and worked as Associate Professor at Department of Computer Science, Nowrosjee Wadia College afflilated to Savitribai Phule Pune University, Pune for last 27 years. Her research interests include Soft Computing, Big Data Analytics, Software Engineering and Cloud Computing.