Journal of the Nigerian Association of Mathematical Physics
Volume34,(March, 2016), pp491 – 496
© J. of NAMP

Improved Resources Utilization Model For Cloud Computing

[1]Oyebode E.O., [2]Ojesanmi O.A. and [3]Folorunso O.

[1]Department of Computer Science, AjayiCrowther University, Oyo, Oyo State
[2]Department of Computer Science, Federal University of Agriculture, Abeokuta, Ogun State
[3]Department of Computer Science, Federal University of Agriculture, Abeokuta, Ogun State.

Abstract

Cloud computing allows the dynamic scaling of computing resources for applications and users via the internet. These resources are shared among customers using virtualization technology. Using these resources efficiently can save cost and boost service delivery. Since cloud computing environment has heterogeneous resources and costs associated with it, testing applications and resource allocation policies are highly challenging. In this study, the effect of virtual machine capacity and various requirements of the applications have been carried out. The relationship among the cloud components such as cloudlet, virtual machine, data center, host etc has been investigated and the improved technique of allocating the cloud resource has been proposed. The proposed technique showed significant improvement for resource utilization and reduction of violation of service level objectives.

Key words: Virtualization, cloud, internet, cloudlet, service level objectives

1.0 Introduction

Cloud computing platforms are rapidly emerging as the preferred option for hosting applications in many business contexts[1]. Cloud computing represents a new way to deploy computing technology so as to give users the ability to access work on, share and store information using the internet. It is an extension of internet that enables services to be made available at reduced cost. Cloud represents the pool of resources with very high processing, memory and networking capabilities [2]. Cloud computing services can be provided to both enterprise and personal users. The cloud technology delivers IT infrastructure and services on demand on pay-as-you-use basis. The use of such a billing scheme makes clouds accessible to everyone, from large academic institutions to minor research groups etc. In addition, Cloud resources are able to scale with respect to users' requests or trade-offs between cost and performance [3]. Virtualization in cloud computing is able to overcome many of the challenges previously identified via the use of the service platforms such as grid. E.g infrastructure as a service solution allows scientists to prepackage and configure the basic building blocks required for carrying out their experiments. This allows a higher degree of customization that helps cover a wider range of scenarios for scientific computing applications. Cloud computing also provides easy integration through provisioning of on-demand cloud resources that can be integrated into existing infrastructures [4].The resource for cloud computing is made available at the data centers for users across different platforms. Scheduling, allocating and managing the resources within the cloud platform constitute challenges [5]. As the cloud expands, quality of service (QoS) control or guarantee grows in scale, the number of users and resources require proper management and allocation. Virtualization itself represents both an advantage (it provides the necessary abstraction to unify fabric components into pool of resources and resource overlays) as well as a disadvantage. Virtualization platforms do not guarantee effective service delivery. A resource management is a concrete process of managing, resource discovery, scheduling, allocation and system workloads. But resource provisioning of a large distributed system (such as a large enterprise web application or the cloud management system itself) is a challenging task [1]. IT administrators of such large distributed system need to constantly manage its capacity to avoid SLA violations as the workload supported by such a system is often very dynamic. The dynamic demand for resources, complexity of enterprise applications and management systems, coupled with heterogeneity of resources in a cloud environment and their non-linear pricing result in many challenging distributed systems and resource management problems [6,7]. Even though, data centres may contain thousands of physical machines able to host tens of thousands of virtual machines each, a poor resource management may put pressure in the data centre capacity thus, data centre resource could be oversubscribed,

Corresponding author: Oyebode E.O., E-mail:kunlebest4u@yahoo.com, Tel:+2348056052007(O.O.A)

underutilized or over-provisioning, resulting in variation in the performance measure. In such scenario, elastic applications hosted in such a stressed data centre may not be able to scale up and this may result in SLA violations [8].

2.0 Literature Review

Most cloud providers operate through data centres that do not adequately utilize the resource [9]. Some recent approaches assigned managers to provide resources to various workloads in a manner that achieves their performance goals [10]. The managers each time need to decide the right amount of resources for each workload and specific servers that can satisfy the resource assignment. The task of determining the resources needed for each user can be very high as the load of each user receiving service(s) can be dynamic [11]. This is why most managers side step the allocation of resources but requesting users or workloads to express their requirements of complex codebases or the variations in load and dataset size [12]. Resource provisioning in the cloud can be classified as used by the cloud providers as statically provisioned, fully reserved system and on-demand instances [13]. Reserved instances require a high upfront capital investment but lower cost when compared to on demand resource provisioning. On demand provisioning is charged on pay as you go basis but can experience violation of service level agreement due to fluctuating workloads [14]. In solving the above problem, various approaches adopted require a breakdown of task qualities that requires voluminous information and partly resource consumption [8]. Other approach requires that resource should be kept unused ahead permanently [15]. This is called statistically reserved resources. This reservation technique keeps resource permanently available for extended period of time. E. g shortest contract for reserved resources on Amazon EC2 is one year. The statically resource provisioning requires that the resources are reserved at once in advance. In other to satisfy the full system load with variability, the cluster is made to provision for the peak requirements of each scenario plus certain amount of over-provisioning. It is important to note that statically provisioning is straight forward when load is static, but when load varies, it results significantly to underutilization of resources [10].

3.0 Proposed Resource Allocation Technique

The application services hosted in cloud have complex provisioning, due to composition issues, configuration setup and varying deployment requirements [5]. To evaluate the performance of such systems using real life system can be very costly in terms of resources and time. To overcome this challenge, CloudSim: an extensible simulation toolkit has been developed. The CloudSim toolkit covers both system and behavior modeling of cloud system components such as data centers, virtual machines (VM), hosts cloudlets etc in a manner similar to real world cloud. CloudSim supports modeling and simulation of cloud computing environments consisting of single and inter-networked clouds (cloud federation) [8]. The relationship among the components of CloudSim in terms of operation is in line with system basic operation. In other to propose a technique of resource sharing we setup a simulation to establish the relationship between the resources (Machine capacities), tasks (applications) and completion times. A graph of task execution time alongside with tasks is shown in Figure 1.

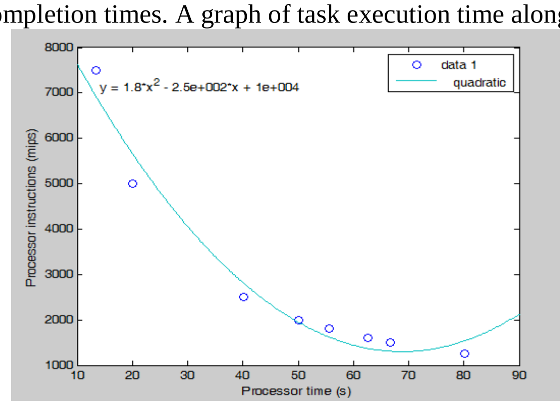

Figure 1: Machine capacities and task completion times

Using CloudSim components to model the interaction between applications and machine capacities, the graph showed a decline in execution time as the machine capacity increases with constant tasks. The study investigated the relationship between the major components of the cloud that serves as resource which is the millions of instructions that can be executed per seconds within virtual machines with respect to host capacity, we then proposed a scheme that depends on the task attributes and virtual machine attributes to allocate the resource within the cloud. The mathematical description of the model is as follows.

A set of tasks submitted at any given instance may be expressed as

$$\propto = \{t_1, t_2, t_3, \ldots, t_n\} \quad (1)$$

Suppose that a set of criteria identified inform of attributes of tasks such as file size, output size, utilization is defined as;

$$\beta = \{c_1, c_2, c_3, \ldots, c_n\} \quad (2)$$

The set of tasks is examined to obtain the weight vector of each task with respect to attributes selected as objective. Computation of pair-wise comparison is computed for each task using n attributes to define n tasks as

$$PC^{j,k} = \begin{cases} \frac{1}{PC^{j,k}}, j \neq k \\ 1 \;\; j = k \end{cases} \qquad (3)$$

$$\begin{bmatrix} \frac{w_j}{w_1}, & \cdots & \frac{w_j}{w_n}, \\ \vdots & \ddots & \vdots \\ \frac{w_n}{w_1} & \cdots & \frac{w_i}{w_n} \end{bmatrix} \begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix} = \lambda_{max} \left[\begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix} \right] \qquad (4)$$

From the matrices, a weight vector of criteria can be obtained by solving (5)

$$PC * \eta^{criteria} = \lambda_{max} * \eta^{criteria} \qquad (5)$$

λ_{max} is the principal Eigenvalue of PC.

Consistency can be checked as follows:

$$PC = \frac{w_j}{w_k} \quad j, k = 1,\ldots,n \text{ \{if and only if it is consistent\}} \qquad (6)$$

Consistency is obtained if

$\eta^{criteria} > 0$ within a multiplicative constant.

$$\text{Consistency ratio, CR} = \frac{CI}{RI} \qquad (7)$$

where CI = $\frac{\lambda_{max} - n}{n}$

RI is the random index obtained from comparison matrix.

Since the virtual machines at the data centre is finite, its attribute can be also be used to form weight vector such that

$$V_w = \{v_1, v_2, \ldots, v_n\} \qquad (8)$$

Scores can be computed between equation (5) and equation (8) for decision making on tasks that can be executed within minimum time and the status of the virtual machine can be made to change state within minimum time.

In the experiment, tasks attributes used include output size, file-size and utilization in terms of sharing resources. While statically scheduled virtual machines were identified and Millions of instructions (MIPS), random access memory (RAM) and image size were used to obtain weight of vectors. The algorithm used for computation of pairwise computation is shown in Figure 2.

```
Input     Vms, cloudlets, intv
Output   vm
t= 0
n= count(cloudlets)
While(t<intv){
//register all cloudlets
Arraylist<Cloudlet> list = new arrayList<Cloudlet>
For(i=0;i<n;i++){
                    list.add(new cloudlet("size", sla)
Compute pcm(list, intv)
}
// Use pcm to scale cloudlet features
Public static T pcm(list cloudletlist, intpcm) {
Obtain cloudlet from cloudlet list
double[][] pcmlab
M = cloudletlist.length
N = pcm
Cloudlet cloudlet = new Cloudlet()
Pcmlab = new double[M][N]
          for(inti=0;i<M;i++){
                    for(int j=0;j<N;j++)
                              pcmlab[i][j] = cloudlet[i].feature/cloudlet[i].feature +cloudlet.feature[j]
          cloudlet.data = pcmlab[i][j]
}
For(inti =0;i<n;i++){
Initial = 0;
If ( initial ==0 ) {
Compval = cloudlet.data
```

ElseifCompval>initial
Initial = cloudlet.data
return cloudlet
}

Figure 2: Algorithm for computing pair-wise comparison

4.0 Results and Discussion

A single cloud data center with resource was modeled with specific objective of assigning Vms to tasks to maximize execution of tasks. The Vms observed is assumed to operate on a single host and single core without parallel execution of Vms. Various sizes of applications were setup in form cloudlets on periodic basis and each were allocated resource in form of virtual machines base on attributes represented by the requests. As it were in real machine world, the capacities of the Vm were also not the same. The technique of allowing dynamic execution was also carried out. The algorithm used adopted the pair-wise comparison to rank features that were used in redistributing the cloudlets to Vms. Outputs of results obtained are shown in figure 3 and figure 4 respectively. The results showed that the proposed model demonstrated improved performance in terms of processing of more cloudlets in the dynamic approach than static approach. More tasks were executed as the resource allocated considered the feature of tasks ahead allocation. Therefore the throughput was increased to about 12%. However, the proposed scheme required additional processing delay before the initial startup of most executions as it took additional time to examine requests before allocation and then execution.

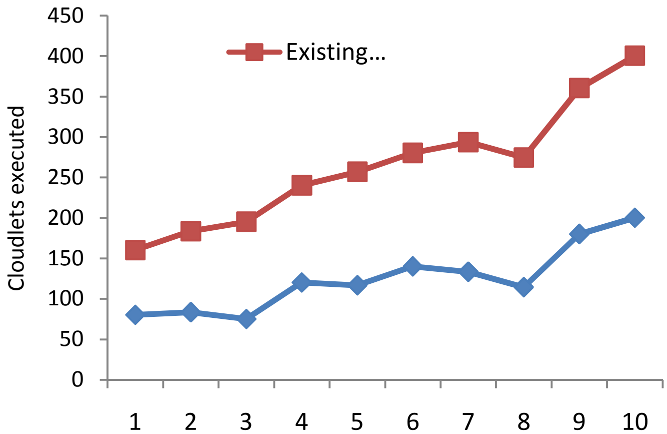

Figure 3: Cloudlets executed against number of Vms

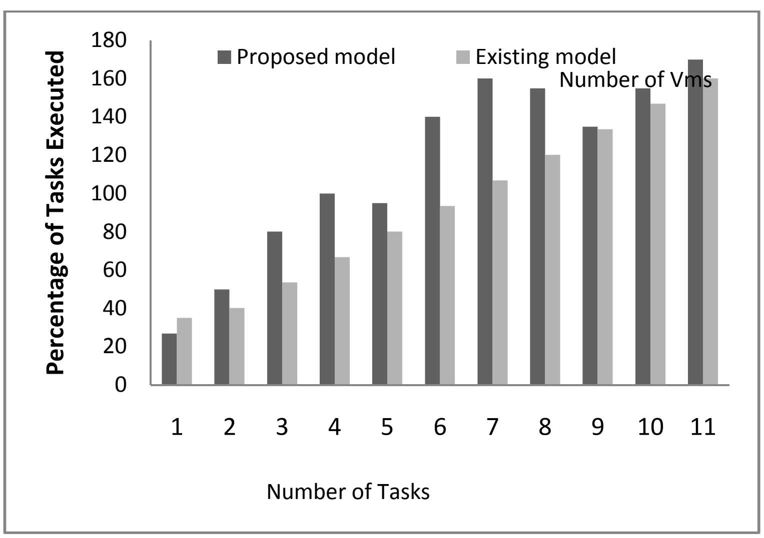

Figure 4: Percentage of tasks executed in data center at different times.

5.0 Conclusion

This study has focused on using heuristics from tasks to allocate resource in the data centre and it has shown improvement in terms of tasks that can be accomplished within some interval of time. the effect of virtual machine capacity and various requirements of the applications have been carried out. The relationship among the cloud components investigated and the improved technique of allocating the cloud resource has been proposed. Further studies examine the policies used in the datacentre to utilize the allocated resources during execution within and outside a single cloud data centre. The technique showed significant improvement for resource utilization in cloud computing.

6.0 Reference

[1] Carrera D., Steinder M., Whalley I., Torres J. and Ayguade E. (2012) Autonomic placement of mixed batch and transactional workloads. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*; **23**(2), pp. 219–231.

[2] Chen H., Wang F. and Helian F. (2013). A Cost-Efficient and Reliable Resource Allocation Model Based on Cellular Automaton Entropy for Cloud Project Scheduling. International Journal of Advanced Computer Science and Applications, Vol. 4(4), pp. 7-14.

[3] Wu, L., Garg, S. K., and Buyya, R. (2011). Sla-based resource allocation for software as a service provider (saas) in cloud computing environments. In Cluster, Cloud and Grid Computing (CCGrid), 11th IEEE/ACM International Symposium on, pp. 195–204.

[4] Son, S., Jung, G., and Jun, S. C. (2013). Ansla-based cloud computing that facilitates resource allocation inthe distributed data centers of a cloud provider. The Journal of Supercomputing, pp. 1–32.

[5] Mollamotalebi M., Maghami R. and Ismail A. S. (2013). Grid and Cloud Computing Simulation Tools, International Journal of Networks and Communications 3(2), pp. 45-52.

[6] Lee E. K., Kulkarni I., Pompili D. and Parashar M. (2012). Proactive thermal management in green datacenters. The Journal of Supercomputing, Vol. 60, No. 2, pp. 165–195.

[7] Ousterhout K., Wendell P., Zaharia M. and Stoica I. (2013). Sparrow:Distributed, Low Latency Scheduling. ACM Symposium on Operating Systems Principles (SOSP), pp. 69-84.

[8] Buyya, R., Garg, S. K. and Calheiros, R. N. (2011). SLA-Oriented resource provisioning for Cloud Computing:Challenges, Architecture and Solutions. Proceedings-2011 International Conference on Cloud and Service Computing. pp. 1-10.

[9] Xu, Z., Tu, Y. –C. and Wang, X. (2012). PET: Reducing database energy cost via query optimization. Proceedings of the VLDB Endowment, 5(12), pp. 1954-1957.

[10] Hsien W., Yang C. and Hwang M. (2016). A survey of public auditing for secure data storage in cloud computing. International Journal of Network Security, 18(1), pp. 133-142.

[11] Yang J., Zeng K, Hu H. and Xi H. (2012). Dynamic cluster reconfiguration for energy conservation in computation intensive service. *IEEE Transactions on Computers.* Vol61(10): pp. 1401–1416.

[12] Goiri Í, Berral J, OriolFitó J, Julià F, Nou R, Guitart J, Gavaldà R, Torres J. Energy-efficient and multifaceted resource management for profit-driven virtualized data centers (2012). *Future Generation Computer Systems (FGCS)*,28(5), pp. 718–731.

[13] Ramanathan R. and Latha B. (2014). Resource optimization based on demand in cloud computing. Research Journal of Applied Sciences, Engineering and Technology, (15), pp.1724-1731.

[14] Sobhanadri K. and Revathi A.(2014). Active Resource Provision in Cloud Computing Through Virtualization. International Journal of Innovative Research in Computer and Communication Engineering. Vol. 2, pp. 4248-253.

[15] Yau, S. S. and Yin, Y. (2011). Qos-based service ranking and selection for service-based systems. In Services Computing (SCC), 2011 IEEE International Conference, pp. 56–63