QUALITY OF SERVICE-BASED RESOURCE ALLOCATION FOR WEB CONTENT DELIVERY ON CLOUD COMPUTING INFRASTRUCTURE

OMOTUNDE, AYOKUNLE A[1], IDOWU, SUNDAY[2] KUYORO, 'SHADE [3], AYANKOYA, FOLASADE Y[4], JOSHUA, JONAH V[5], ADEGBENJO, ADERONKE [6], ABEL, SAMUEL B.[7]

[1]Information Technology Department, School of Computing and Engineering Sciences, Babcock University, Ilishan Remo, Ogun State, Nigeria.
[3,5,6,7] Computer Science Department, School of Computing and Engineering Sciences, Babcock University, Ilishan Remo, Ogun State, Nigeria.
[2, 4]Software Engineering Department, School of Computing and Engineering Sciences, Babcock University, Ilishan Remo, Ogun State, Nigeria.

Email: [1]ayo_omotunde@yahoo.com, [2]idowus@babcock.edu.ng, [3]afolashadeng@gmail.com, [4]sadeayankoya@gmail.com, [5]joshuaj@babcock.edu.ng, [6]adegbenjoa@babcock.edu.ng, [7]abels@babcock.edu.ng

ABSTRACT

Demand for web content continues to increase at exponential rates and this has intensified the challenges of satisfying customer's Quality of Service. Several techniques for Web content delivery vis-à-vis resource allocation have been proposed, one of which is the use of Content Distribution Networks. However, in recent times, cloud computing has become a driving force in Information Technology where Service Providers' limited resources are shared among numerous users with different QoS requirements. In this work, focus is on developing a model for allocation of resources on cloud computing Infrastructure in order to improve delivery of Web content and optimize service cost. An analytical approach was adopted and expressed as an optimization problem subject to QoS metrics: delay, throughput, and bandwidth. The optimization problem was formulated as an Integer Linear Programming problem in which the decision variable takes the value of 0 or 1. A single Infrastructure-as-a-Service cloud with Virtual Machine (VM) instances running in Physical Machines (PM) was assumed. The model was considered for different values of delay, throughput, and bandwidth for each VM to obtain minimum cost of delivering Web content to users. An algorithm was developed and sample data were collected from Amazon Elastic Cloud Compute/storage pricing model to obtain optimal results. The implementation of the algorithm was done using 'A Mathematical Programming Language/Modular In-core Nonlinear Optimization Systems' (AMPL/MINOS).

Key words: *Quality of Service, Resource allocation, Web content, Web content delivery, cloud computing*

1. INTRODUCTION

Delivery of Web Contents such as text, images, sounds and video became proliferated with the introduction of the Internet to the public. Driven by rapid acceptance of broadband access along with increase in system complexity and content richness, the Internet has experienced tremendous growth and maturity. In addition, with the introduction of a plethora of Internet enabled devices, the number of users continues to grow at a quick rate and this in turn has altered internet usage pattern from being partially online to always online, thus resulting into more requests for more Web Contents.

As a result of this growth and the pervasiveness of the Internet, there has been an unusually large growth in network traffic generated by request for and response to Web contents. If the traffic grows to an extent where either the server's processing capacity or storage space, the bandwidth available on its connection to the Internet can easily be maxed out, user requests are dropped, which results in increased access delays and less requests being responded to (i.e. lower throughput). Since the dawn of the Internet, efforts have been made to ensure that it not only delivers Web Contents to users but also ensures that these contents are delivered to meet the user's Quality of Service expectations such as higher throughput and minimal delay while request is being responded to. One approach was to modify traditional Web architecture by upgrading the web server hardware, adding a high-end processor, upgrading the memory and adding to the disk space. This approach slightly helped to reduce the performance problem. However, it did not provide a lasting solution because of the fact that traffic increased geometrically as more and more people surf the web on a daily basis. Moreover, this approach of upgrading hardware components was not flexible as opined by [1] because it gets to certain point when small enhancements became impossible and the only option was to replace the entire web server system [2].

An initial solution to the problem of ensuring that Web content were delivered to meet user's QoS expectation was proposed by [3]. This method improved performance, reduced server load and at the same time reduced bandwidth usage, especially for narrowband users by deploying caching proxies that serve user's request. It helped to meet growing demands on the Internet by improving speed, throughput, and availability. Speed was improved by successfully migrating copies of frequently requested documents from the server to a cache closer to the clients. By this, clients experienced shorter delays when requesting for content.

A different approach for better performance is the use of server farms. A server farm is a group of networked servers that distribute tasks in a way that maximizes efficiency and minimizes the risk of losing data. According to [1], each server in the farm shares the burden of responding to requests for the same web site. Although server farms and hierarchical caching (through caching proxies) are useful techniques to address the Internet Web performance problem, they have limitations. In the first case, since servers are deployed near the origin server, they do little to improve the network performance due to network congestion. This may force the content providers with a popular content source to invest in large server farms, load balancing, and high bandwidth connections to keep up with the demand.

To address these limitations, Content Distribution Network or Content Delivery Network (CDN), which is a system of computers networked together across the Internet to cooperate transparently for delivering content to end-users, was developed. It involves a set of surrogate servers (distributed around the world) that cache the origin servers' content; routers and network elements that deliver content to the optimal location and the optimal surrogate server and an accounting mechanism that provides logs and information to the origin servers.

Today, cloud computing offers businesses and content providers an inexpensive way to expand their infrastructure with the use of shared pool of configurable computing resources that may belong to the same or different service providers [4]. Cloud resources can be seen as any resource, be it physical or virtual, that users may request from the Cloud. These include network requirements, storage, computational needs such as CPU time, or even software applications [5]. These resources are usually placed in multi-tenant data centers that are able to match the resources with the volume of work being done at any point in time such that an expansion in business activities leads to more resources being provisioned and a contraction leads to less resources being provisioned.

Though, cloud is defined as both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services, due to the dynamic nature of demand from users, it is not certain that service providers may be able to fully satisfy these demands [6]. This poses a challenge in the areas of quality, availability, usability, and reliability of services provided. Furthermore, efficient delivery of Web Content to users have always been a phenomenon that requires guaranteed quality of service and resource provisioning since the Internet was introduced to the general public. The adoption of Cloud computing has led companies to embrace new and cost-effective ways to harness Information Technology infrastructure. However, delivery of Web content on this new computing paradigm also requires a guaranteed quality of service. Therefore, this work proposes a model for resource allocation for Web content delivery on cloud Computing to improve efficiency subject to quality of service constraints like delay, throughput and bandwidth.

2. REVIEW OF RELATED WORKS

Quite number of comprehensive reviews have been done about this subject area which can be found in the works of [7] [8] [9] [10]. However, it may be necessary to explicate how newer methods have been able to improve over older existing methods.

When provisioning resources to tasks in the Cloud it is possible to have a few idle resources that may be unused at times. [11] proposed and developed a resource allocation mechanism that integrates and allocates these idle resources to users by introducing microeconomic methods into the resource management and allocation in the Cloud environment. By combining batch matching and reverse auction, a reverse batch matching auction mechanism was proposed for resource allocation. Market efficiency, user satisfaction and QoS are maximized in an optimization problem to determine the winner of the auction. [12] also proposed an auction mechanism that worked well under certain conditions such as when a value is specified for the bandwidth between the Cloud user and servers. Cloud servers are allotted to users by an auction mechanism that checks whether the unused bandwidth of a server is greater than a specified value. This method, however, did not consider the cost of executing specific tasks on the available server. Furthermore, [13] proposed another auction-based mechanism that helps the cloud service provider to decide how and when to allocate resources and to which users. The mechanism is most useful in real-time environment and can give a near-optimal solution. It is, however, not practical when the resources and tasks are known ahead and are required to be scheduled to optimize utilization of the system. [14] proposed an increase in Cloud service provider profit by reducing the penalty cost for Service Level Agreement violation. Execution time as SLA constraint in combinatorial auction system was considered where there are series of bidding rounds in which winners are determined according to job's urgency. The result, at the end of each bidding round, is used to efficiently allocate resources and reduce penalty cost.

Two provisioning plans for computing resources are mainly in use. Reservation plan and On-demand plan. [15] proposed the Robust Cloud Resource Provisioning (RCRP) algorithm to achieve the best advance reservation. The RCRP came as an improvement on the existing work of [16] Optimal Cloud Resource Provisioning (OCRP) that used two uncertainties (demand and price) to find an optimal solution for resource provisioning and VM placement. The RCRP considers four uncertainties (demand, profit, resource utilization and cost uncertainty) to get a more robust solution.

Leveraging on Toyota's Just-in-Time philosophy, [17] were able to address the problems that emanate from capacity planning in the Cloud. For efficient provisioning of cloud data centers, computational infrastructures of a cloud computing provider are assembled based on the costs that have already been absorbed by the core businesses that use them. This resource allocation strategy ensures that the provider allocates resources only when demanded and until there is use for them. Built upon the amortized resources from a supply chain, JiT Clouds may represent an attractive alternative for many types of clients and applications both in price and in scalability. Amortized resources are gotten as a result of a federation of low scale resources already existing. Just in Time Provider is a public cloud computing provider that makes use of a federation of low scale amortized resources already existing into private contexts instead of assembling and maintaining a structure of data centers for supporting its own services. Unlike proxies of conventional providers of cloud computing, a Just in Time Provider does not represent any public cloud provider, but acts as a legitimate and fully autonomous provider that takes advantage of resources that would be irretrievably wasted without its intervention.

Topology based resource allocation was proposed by [18] in which an architecture that gathers information about hosted application requirements without the explicit user input. This information is used to forecast the performance of a particular resource allocation. This architecture is referred to as TARA and it is made up of a prediction engine that uses a lightweight simulator to estimate the performance of a given resource allocation and a search engine that makes use of genetic algorithm to find an optimal solution in a large search space.

A number of mechanisms based on genetic algorithm were propose, one of which was presented by [19] for task level scheduling in Hadoop MapReduce. The major advantage of this work is that it could help find the local optimum solution, however the execution of the load balancing algorithm may take a long time to make a decision for the task assignment thus impacting the overall performance. Another genetic-based mechanism was presented by [20] in order to minimize the waiting time of tasks to be scheduled in a cloud computing environment. Furthermore, [21] addressed the independent batch scheduling in computational grid by presenting a genetic based algorithm in order to solve the global minimization problem in grid-based energy consumption. The main disadvantage of this work, is that it is based only on two criteria, while fixing several other parameters.

The challenge of providing and ensuring QoS for mobile users in Cloud computing environment was addressed in the work of [22] by developing a QoS framework for mobile computing and adaptive QoS management process to manage QoS assurance in mobile computing environment. In addition, a QoS management model based on Fuzzy Cognitive Map (FCM) was proposed. This work facilitated QoS prediction, establishment, and assessment. The problem with the work is that a good model with suitable configurations was not generated.

[23] proposed a QoS-Aware Dynamic Data Replica Deletion Strategy (QDDRDS) for disk space and maintenance cost saving purposes because distributed storage systems which forms the foundation of all kinds of services provisioned in the Cloud is the underlying infrastructure of Cloud Computing. These distributed storage systems hold replica which enhance the reliability and performance of the system. This reliability comes at a cost in terms of disk space and maintenance cost. While the QDDRDS saved disk space and maintenance cost, the availability and performance QoS requirement are ensured. However, there is an increased overhead in terms of updating the distributed storage. Also, there tends to be inconsistency of data reducing the QoS availability.

So far, the works that have been examined tend to focus on resource allocation strategies in the cloud with special attention to work flow with the exception of [22] which was fixated on mobile users. Non seemed to focus on QoS. QoS entails techniques for managing network resources since achieving the required QoS by managing the delay, jitter, bandwidth, amongst many other metrics on a network holds the key to successful end-to-end business solutions. In Cloud Computing, the issue of QoS is a pertinent issue since cloud users expect their service providers to provide resources in a way that it meets the advertised characteristic and the expectations stipulated in the SLA. This is not an easy task since cloud providers need to find the right tradeoff between operational costs and QoS levels.

A video-conferencing system was used to propose a resource allocation model for QoS management that allocates resources to different applications in order to satisfy various quality of service requirements by [34]. The QoS-based Resource Allocation Model (Q-RAM) as it was referred to, assumed a system with multiple concurrent applications, each having its own set of QoS level based on the system resource available to it. The objective of the Q-RAM is to make resource allocations to each application such that the total system utility is maximized under the constraints that every application is feasible with respect to each QoS dimensions. The total system utility which is to be maximized is an aggregate of all application utility. Hence [35] proposed a scheduling algorithm referred to as Multiple QoS constrained scheduling strategy for multiple workflows (MQMW) to address the challenge that comes up as a result of the uniqueness of each request in terms of QoS requirements from these numerous clients. Workflows which can be started at any time with different QoS requirements are scheduled upon arrival with high success rate. Results of the experiments conducted on this algorithm yielded better scheduling results. However, QoS constraints such as availability and reliability were not considered.

A Mixed Integer Non-Linear Programming problem was formulated by [24] to solve the problem of task planning. Their model assumed multiple heterogeneous compute and storage cloud provider and parameterized them by cost and performance in addition to the constraints on the maximum number of resources on each cloud. The objective function in this work minimized total cost of work flow execution under deadline constraints. This work however differs from the focus of this paper because it focused on optimizing tasks and work flow while the focus in this paper is on QoS based delivery of Web Content.

There is a challenge of incorporating a comprehensive QoS demand for Big data with cloud computing while minimizing total cost. As a solution to this challenge [25] proposed heuristic algorithms which were implemented based on the premise that the reduction of resource waste has a direct relation on minimization of cost. These algorithms are equipped with tuning parameters to find minimized dynamic resource allocation solutions but it doesn't consider metrics such as delay, jitter (delay variation) and throughput.

While a number of researchers have tried to solve problems in other optimization areas like resource scheduling and workflow little or no work, has been done explicitly to minimize total service cost incurred in transferring web objects from service provider's infrastructure to clients in the client network subject to specific QoS constraints like delay, throughput and bandwidth.

3. METHODOLOGY

3.1 Problem Formulation

Consider a Cloud provider and clients whose processes are described by [26] as follows:

1. Client in the client's network generates request through the network to compute cloud.
2. VM on compute cloud gets data from storage cloud
3. VM on compute cloud aggregates information for client
4. VM on compute cloud responds with information to client through network.

The location of the storage clouds is not specified in 2 above. However, in this work, it is assumed that there are multiple storage locations holding replicas of the requested Web Objects as we have in Content Delivery Networks (CDN).

From the processes described, three costs are identified. These are i) Cost of VM. ii) Cost of storage (replicated in multiple storage servers across the globe (iii) Cost of transfer of data from storage to client.

A Cloud provider has physical machine that may hold instances of virtual machines and storage.

The input sets for the formulation of the problem are as follows:

I = A set of physical machines.
J = A set of virtual machines in physical machine.
N = A set of nodes in the client network.
K = A set of web objects or content
S = A set of storage units

The parameters that describe a virtual machine are as follows.

cj = cost of a virtual machine j, for each j in J
vj = number of virtual machine instances j in J
tj = number of hours used by virtual machine, for each j in J
min_bj = minimum bandwidth required by virtual machine j, for each j in J
max_bj = maximum bandwidth required by virtual machine j, for each j in J
min_tj = minimum throughput required by virtual machine, for each j in J
max_tj = maximum throughput required by virtual machine, for each j in J
sj = storage available to virtual machine j, for each j in J

The parameters that describe a web object are as follows.

sk = size of object k
Σ sk = aggregate size of all objects k (for all objects)
rk = request rate for object k by client n
δk = unit delay for transferring object k
Δk = δkΣ rk Aggregate delay for all k (for all objects)
λk = sk / δk Throughput for transferring object k
bk = bandwidth for transferring object k
β = Σ bk
ck = cost of outbound transfer of object k
ct = sk . rk . ck Cost of transferring object k from storage to client n.

Parameters that describe storage cs = cost of storage/unit

Scap = storage capacity. The cost of storage is taken into consideration because there are certain instances of VM that don't have within them storage units.

QoS metrics considered in the SLA

t = throughput as stated in the SLA
b = bandwidth as stated in the SLA
d = delay as stated in the SLA

Xij = 0 or 1 defines the decision variable representing the virtual machine j in physical machine i responding to request from client n in N. The decision variable takes the value 1 if the virtual machine serves the request otherwise the value is 0.

The cost of transferring a single web object k includes the cost of Virtual M instance, cost of transferring the object from storage to client n, and the cost of storage. This can be described as:

cj + sk.rk.ck + sc (i)

The aggregate cost of all virtual machines j in all physical machines i is as follows:

$$\sum_{i\in I}\sum_{j\in J} C_j v_j t_j X_{ij} \qquad \text{(ii)}$$

The cost of transferring all objects k in all virtual machine j in all physical machines is as follows:

$$\sum_{i\in I}\sum_{j\in J}\sum_{k\in K} S_k r_k C_k X_{ij} \qquad \text{(iii)}$$

Cost of storage in all storage sites around globe

$$\sum_{s\in S} s_c \qquad \text{(iv)}$$

The objective function therefore represents the total cost of transferring multiple web objects from storage to client n in N. It is defined as:

$$Z = min \sum_{i\in I}\sum_{j\in J} C_j v_j t_j X_{ij} + \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} S_k r_k C_k X_{ij} + \sum_{s\in S} s_c \qquad \text{(v)}$$

for all object k in all virtual machine j, in all physical machine i.

subject to:

min_bj < Σ bk < max_bj (1)
min_tj > λk ≤ max_tj (2)
Δk ≤ d (3)
Σsk ≤ Scap (4)
Xij = 0 or 1 (5)

Interpretation of Constraints

(1) enforces that the aggregate bandwidth consumed while transferring Web objects fall between the minimum and maximum bandwidth stipulated in the Service Level Agreement.

(2) guarantees that the actual throughput for transferring Web objects falls within the range of what is bargained for by clients

(3) imposes that the aggregate delay for all the Web objects is less than or equal to the delay stated in Service Level Agreement.
(4) ensures that the storage capacity of VMs are not exhausted
(5) says that if virtual machine j is used to transfer object k from storage to client, the decision variable takes the value of 1, otherwise it is 0.

4. EXPERIMENTAL CONSIDERATIONS

4.1 Workspace and Equipment

Experiments were carried out on a system with the following specification:

- Intel Core i3-4005U, 1.7 Dual Core, 4GB RAM, 500GB HDD. One was done considering a single storage unit.

The total service cost for delivering Web content was obtained when 2, 4, 6, 8 and 10 virtual machines were instantiated respectively in 1 physical machine for varying number of Web objects. The number of physical machines was then increased to 2 and 2, 4, 6, 8, 10 virtual machines were instantiated respectively for varying number of Web contents. Lastly, the number of physical machines was increased to 3 while 2, 4, 6, 8, and 10 virtual machines were instantiated respectively for varying number of Web contents.

4.2 Illustrative Examples

The study considered number of (PM,VM,Object) = {(1,2,2), (1,2,4), (1,2,6), (1,2,8), (1,2,10)}, num(PM,VM,Object) = (1,6,2), (1,6,4), (1,6,6), (1,6,8), (1,6,10)}, num(PM,VM,Object) = {(1,10,2), (1,10,4), (1,10,6), (1,10,8), (1,10,10)}.

For 2 PMs, num(PM, VM, Object) = {(2,2,2),(2,2,4),(2,2,6),(2,2,8),(2,2,10)}, num(PM,VM,Object) = {(2,6,2), (2,6,4), (2,6,6), (2,6,8), (2,6,10)}, num(PM,VM,Object) = {(2,10,2), (2,10,4), (2,10,6), (2,10,8), (2,10,10)}.

For 3 PMs, num(PM,VM,Object) = {(3,2,2), (3,2,4), (3,2,6), (3,2,8), (3,2,10)}, num(PM,VM,Object) = {(3,6,2), (3,6,4), (3,6,6), (3,6,8), (3,6,10)}, num(PM,VM,Object) = {(3,10,2), (3,10,4), (3,10,6) ,(3,10,8), (3,10,10)}.

The service costs were obtained for all the cases considered in the methodology and the result indicated the configuration that yielded minimum service cost. The most profitable configuration for transferring 2, 4, and 6 objects occurred when the number of PM was 1 and VM was 2. For 8 objects, the most profitable configuration was when the number PM was 2 and VM was 2. Lastly, for 10 objects, the most profitable configuration occurred when the number of PM was 2 and VM was 2. This indicates that the model was effective in allocating resources that will result in minimizing the total service cost associated with the transfer of Web objects.

4.3 Results and Discussion

In this section, the performance of the resource allocation model proposed in the methodology was studied by presenting the total service cost for all the simulations that were stated in the Illustrative example in Table 1. The results indicated the configuration of physical machine and virtual machine that yielded minimum service cost. The most profitable configuration for transferring 2, 4, and 6 objects occurred when the number of physical machines was 1 and virtual machine was 2. For 8 and 10 objects, the most profitable configuration occurred when the number of physical machines was 2 and virtual machine was 2. This result is an indication that the model was effective in allocating resources that will result in minimizing the total cost associated with the transfer of Web content.

Figures 1 to 9 are extracted from the information on Table 1 and they are more precise as they each focus on peculiar instances. With reference to Figure 1 to Figure 9, the total cost of service reduced as the number of Web objects increase. This indicates that the model was effective in minimizing the total service cost associated with the transfer of Web objects.

Figures 10 to 12 indicated that the total cost of service increased for static number of objects and increasing number of Virtual Machines. This implies that increasing the number of Virtual Machines must be justified by the availability of bandwidth and throughput. In other words, it is not cost effective to increase the number of virtual machine if the bandwidth on currently running instances have not been utilized to the maximum.

The results presented in Table 1 also indicated the most profitable configuration for transferring Web content that will result in least service cost. The most profitable configuration for transferring 2, 4 and 6 Web objects occurred when the number of physical machines was 1 and the number of virtual machines was 2. For 8 objects, the most profitable configuration occurred when the number of physical machines was 2 and the number of virtual machines was 2. Lastly, for 10 objects, the most profitable configuration occurred when the number of physical machines was 2 and the number of virtual machines was 2. These results indicate that the resource allocation model that was developed in this work was effective in minimizing total service cost associated with the transfer of Web objects and helps to choose the best configuration that will yield minimum total service cost.

Table 1: Total Service Cost for transferring Web Content

PM	VM	No of Objects	Total Service Cost ($)
1	2	2	0.647
		4	0.38
		6	0.202
		8	0.2
		10	0.19
	6	2	0.673
		4	0.559
		6	0.329
		8	0.293
		10	0.25
	10	2	1.948
		4	1.521
		6	0.758
		8	0.659
		10	0.619
2	2	2	5.215
		4	0.541
		6	0.297
		8	0.29
		10	0.1
	6	2	1.071
		4	0.842
		6	0.396
		8	0.352
		10	0.165
	10	2	2.711
		4	2.054
		6	0.926
		8	0.799
		10	0.45
3	2	2	1.048
		4	0.599
		6	0.329
		8	0.292
		10	0.2
	6	2	1.717
		4	1.352
		6	0.741
		8	0.659
		10	0.618
	10	2	2.889
		4	2.211
		6	1.035
		8	0.899
		10	0.842

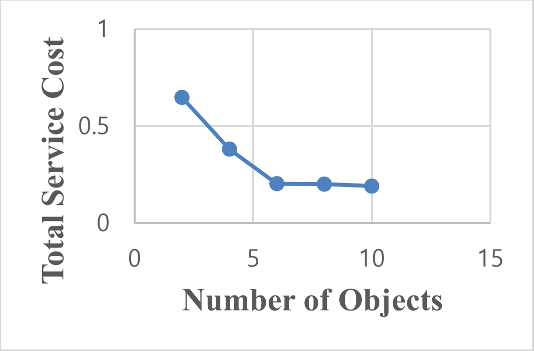

Figure 1 Total cost for 1 Physical Machine, 3 Virtual Machines and varying number of objects

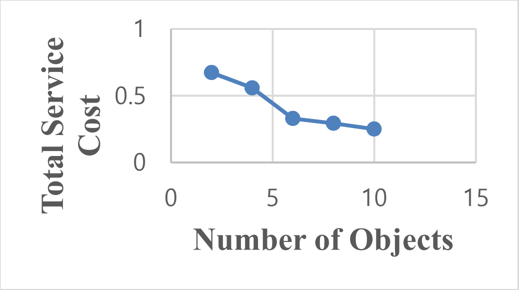

Figure. 2 Total cost for 1 Physical Machine, 6 Virtual Machines and varying number of objects

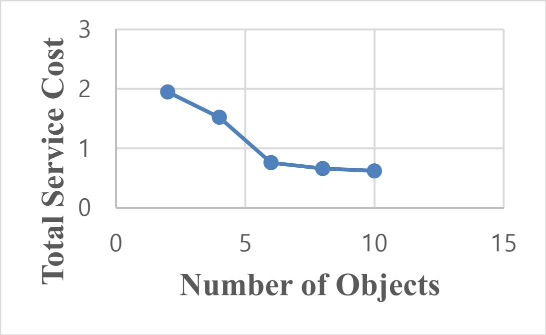

Figure 3 Total cost for 1 Physical Machine, 10 Virtual Machines and varying number of objects

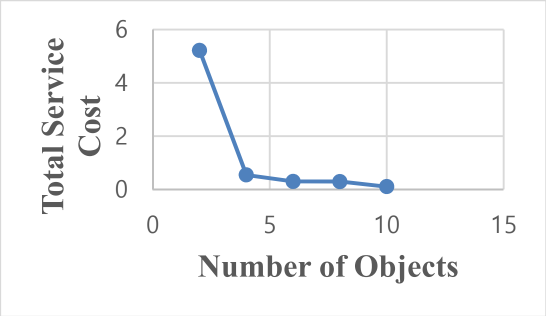

Figure. 4 Total cost for 2 Physical Machine, 2 Virtual Machines and varying number of objects

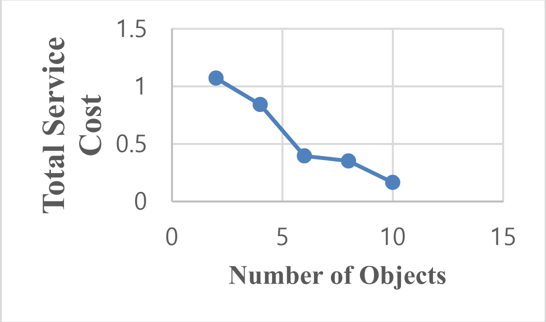

Figure. 5 Total cost for 2 Physical Machine, 6 Virtual Machines and varying number of objects

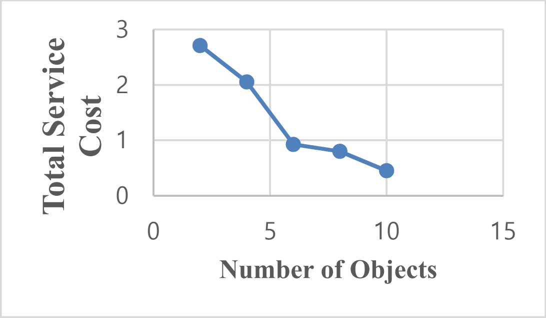

Figure. 6 Total cost for 2 Physical Machine, 10 Virtual Machines and varying number of objects

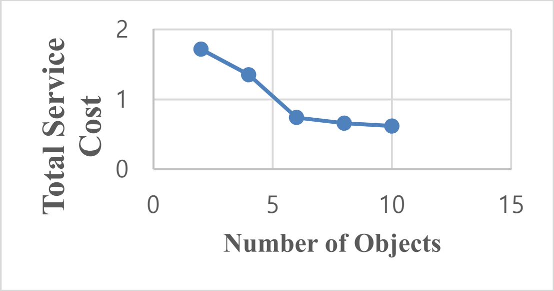

Figure. 7 Total cost for 3 Physical Machine, 2 Virtual Machines and varying number of objects

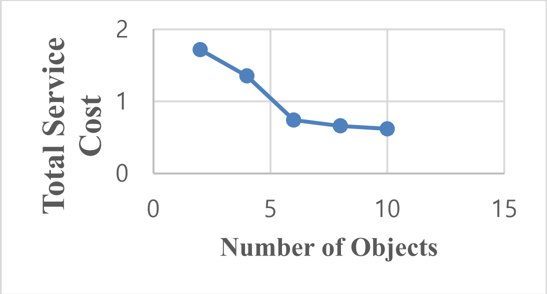

Figure. 8 Total cost for 3 Physical Machine, 6 Virtual Machines and varying number of objects

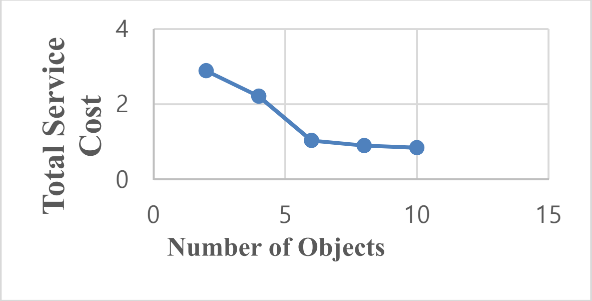

Figure. 9 Total cost for 3 Physical Machine, 10 Virtual Machines and varying number of objects

Table 2 Total cost for static number of Web objects and increasing number of VMs on 1 PM

PM	VM	O	Total Service Cost ($)
1	2	2	0.647
1	6	2	0.673
1	10	2	1.948

Table 3 Total cost for static number of Web objects and increasing number of VMs on 2PM

PM	VM	O	Total Service Cost ($)
2	2	2	5.215
2	6	2	1.071
2	10	2	2.711

Table 4: Total cost for static number of Web objects and increasing number of VMs

PM	VM	O	Total Service Cost ($)
3	2	2	1.048
3	6	2	1.171
3	10	2	2.889

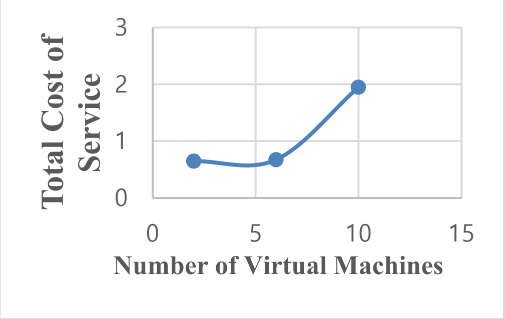

Figure. 10 Static Number of Web Object, 1 Physical Machine, and varying number of VM

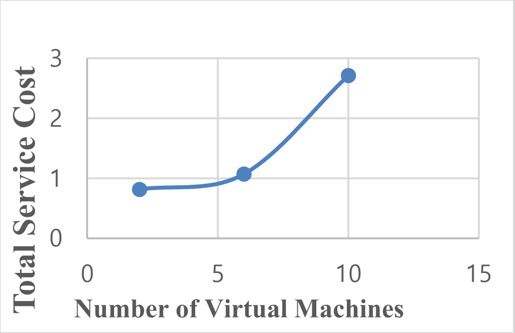

Figure. 11 Static number of Web Objects, 2 physical machine, and varying number of VM

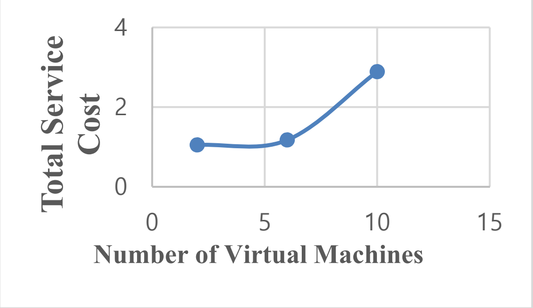

Figure. 12 Static Number of Web Object, Physical Machines, and varying number of VM

The results obtained in this work solve the problem of cost associated with the transfer of Web content. This work also contributes to the body of knowledge as it extends a solution to the resource allocation problems for Web content delivery in cloud computing. The model is recommended for cloud service provides to enable them know how much resources to provision for efficient deliver and optimized cost. This work however focused on a single Infrastructure-as-a-Service Cloud with virtual machines running in physical machines. As a suggestion for further studies, multiple Infrastructure-as-a-Service cloud could be considered.

5. CONCLUSION

Cloud computing systems are getting more complex and the need to satisfy multiple users who demand cloud services has always been an area that requires close attention. providing optimized solutions for scheduling services using a limited number of resources is problem that has gained attention due its impact on cloud computing services.

This study reviewed quite a number of works which include [27], [28], [29], [14], [30] have proposed ways of allocating resources in the Cloud for efficiency. However, these works considered parameters like execution time of tasks, completion time of computing resources, and energy consumption for efficiency. [24] and [25] however, minimized total cost of work flow execution under deadline constraints. They did not consider total service cost for transferring Web contents to requesting users. Thus, in this work, a model for allocation of resources on the Cloud with multiple Web storage was conceptualized, developed and implemented for solving problems of optimizing resource allocation and cost minimization subject to QoS constraints.

REFERENCES

[1]. M. Hofmann and L. R. Beaumont, Content Networking: Architecture, Protocols, and Practice, San Francisco, CA, USA: Morgan Kaufmann Publishers, 2005.

[2]. A. Pathan and R. Buyya, "A Taxonomy and Survey of Content Delivery Networks," 2008.

[3]. M. Abrams, C. R. Standridge, G. Abdulla, S. Williams and E. A. Fox, "Caching Proxies: Limitations and Potentials," 17 July 1995. [Online]. Available: http://www.w3.org/Conferences/WWW4/Papers/155/#KWAN94. [Accessed 27 December 2013].

[4]. B. Furht and A. Escalante, Handbook of Cloud Computing, Springer, 2010.

[5]. G. E. Goncalves, P. T. Endo, T. C. Cordeiro, A. V. Palhares, D. Sadok, J. Kelner, B. Melander and J. Mangs, "Resource Allocation in Clouds: Concepts, Tools and Research Challenges," in *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 2011.

[6]. M. Armbusrt, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Bekeley, 2009.

[7]. V. Vinothina, R. Sridaran and P. Ganapathi, "A Survey of resource allocation strategies in Cloud computing," *International Journal of Advanced Computer Science and Applications,* vol. 3, no. 6, pp. 97-104, 2012.

[8]. M. S. Sagar, B. Singh and W. Ahmad, "A Survey on cloud computing resource allocation strategies," *International Journal of Advance Research and Innovation,* no. 1, pp. 107-114, 2013.

[9]. S. Khan, "A survey on scheduling-based resource allocation in cloud computing," *Journal for Technological Research in Engineering,* vol. 1, no. 1, 2013.

[10]. A. A. Omotunde, S. O. Okolie, S. A. Idowu and O. B. Ajayi, "Resource Allocation in Cloud Computing - An Expose," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 5, no. 9, pp. 543 - 551, 2015.

[11]. X. Wang, Sun, H. Li, C. Wu and M. Huang, "A Reverse Auction Based Allocation Mechanism in the Cloud Computing Environment," *International Journal of Applied Mathematics and Information Sciences,* vol. 7, no. 1, pp. 75-84, 2013.

[12]. H. Chang, H. C. Lu, Y. H. Huang, Y. W. Lin and Y. J. Tzang, "Novel Auction Mechanism with factor distribution rule for Cloud Resource Allocation," *The Computer Journal,* 2013.

[13]. C. Lee, P. Wang and D. Niyato, "A Real-time Group Auction System for Efficiient Allocation of Cloud Internet Applications," *IEEE Transactions on Services Computing,* vol. 8, no. 2, pp. 251-268, 2015.

[14]. C. Yeongho and Y. Lim, "Optimization Approach for Resource Allocation on Cloud Computing for IoT," *Internation Journal of Distributed Sensor Networks,* 2016.

[15]. L. Meera and L. Mary, "Effective Management of Resource Provisioning Cost in Cloud Computing," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 3, no. 3, pp. 75-78, 2013.

[16]. S. Chaisiri, B. S. Lee and D. Niyato, "Optimization of Resource Provisioning Cost in Cloud," *IEEE Transactions on Services Computing,* vol. 5, no. 2, 2012.

[17]. R. Costa, F. Brasileiro, G. L. de Souza Filho and D. M. Sousa, "Just in Time Clouds: Enabling Highly-Elastic Public Clouds over Low Scale Amortized Resources," Universidade Federal de Campina Grande, 2010.

[18]. G. Lee, N. R. Tolia and R. H. Katz, "Topology Aware Resource Allocation for Data-Intensive Workloads," *ACM SIGCOMM Computer Communication Review,* pp. 120-124, 2010.

[19]. Y. Ge and G. Wei, "GA-based task scheduler for the Cloud Computing Systems," in *International Conference on Web Information Systems and Mining (WISM)*, 2010.

[20]. S. Saha, S. Pal and P. K. Pattnaik, "A Novel scheduling Algorithm for Cloud Compouting Environment," in *Computational Intelligence in Data Mining*, 2016.

[21]. J. Kolodziej, S. U. Khan, L. Wang and A. Y. Zomoya, "Energy efficient genetic-based scheduler in computational grids," *Concurrency and Computation: Practice and Experience,* vol. 27, no. 4, pp. 809-829, 2015.

[22]. P. Zheng and Z. Yan, "A QoS-Aware System for Mobile Cloud Computing," in *In the Proceedings of IEEE International Conference on Cloud Computing and Intelligent Systems*, Beijing, 2011.

[23]. B. Liao, J. Yu, H. Sun and M. Nian, "A QoS-aware Dynamic Data Replica Deletion Strategy for Distributed Storage Systems under Cloud Computing Environments," in *In the Proceedings of the Second International Conference on Cloud and Green Computing*, Xiangtan, 2012.

[24]. M. Maciej, F. Kamil and N. Jarek, "Cost Minimization for Computational Application on Hybrid Cloud Infrastructure," *Future Generation Computer Systems,* pp. 65-73, 2013.

[25]. M. M. Hassan, B. Song, S. M. Hossain and A. Alamri, "QoS-aware Resource Provisioning for Big Data Processing in Cloud Computing Environment," in *In the Proceedings of International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, 2014.

[26]. I. Iyoob, E. Zarifoglu and A. B. Dieker, "Cloud Computing Operations Research," *Service Science,* vol. 5, no. 2, pp. 88-101, June 2013.

[27]. K. Rajwinder and P. Luthra, "Load Balancing in Cloud System using Max Min and Min Min Algorithm," *International Journal of Computer Applications,* 2014.

[28]. A. T. Saraswathi, Y. Kalaashri and S. Padmavathi, "Dynamic Resource Allocation Scheme in Cloud Computing," *Procedia Computer Science,* vol. 47, pp. 30-36, 2015.

[29]. T. Zhuo, Q. Z. C. Ling, L. Kenli, S. U. Khan and L. Kequin, "An Energy-Efficient Task Scheduling Algorithm in DVFS-enabled Cloud Environment," *Journal of Grid Computing,* 2015.

[30]. Z. Qian, G. Yufei, L. Hong and S. Jin, "A Load Balancing Task Scheduling Algorithm based on Feedback Mechanism for Cloud Computing," *International Journal of Grid and Distributed Computing,* vol. 9, no. 4, pp. 41-52, 2016.

[31]. H. A. Akpa and B. R. Vadhanam, "A Survey on Quality of Service in Cloud Computing," *International Journal of Computer Trends and Technology (IJCTI),* vol. 27, no. 1, pp. 58 - 63, 2015.

[32]. V. Anuradha and D. Sumathi, "A survey on resource allocation strategies in cloud computing," in *In International Conference on Information Communication and Embedded Systems (ICICES)*, 2014.

[33]. B. P. Rimal, E. Choi and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," in *Fifth International Joint Conference on INC, IMS, and IDC*, 2009.

[34]. R. Rajkumar, C. Lee, J. P. Lehoczky and D. Siewiorek. "A Resource Allocation Model for QoS Management" *in the Proceedings of the 18th IEEE Real-Time Systems Symposium,* pp. 298 – 307, San Francisco, CA, USA, 1997.

[35]. M. Xu, L. Cui, H. Wang and Y. Bi "A Multiple QoS constrained scheduling strategy of multiple workflows from Cloud Computing" *in the proceedings of IEEE International Symposium on Parallel Distributed Processing,* pp. 629 – 635, Chengdu IEEE, doi: 10.1109/ISPA.2009.95, 2015