Indonesian Journal of Electrical Engineering and Computer Science
Vol. 12, No. 3, December 2018, pp. 1179~1186
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v12.i3.pp1179-1186

An Efficient Framework to Improve QoS of CSP using Enhanced Minimal Resource Optimization based Scheduling Algorithm

Ravi Mahadevan[1], Neelamegam Anbazhagan[2]
[1]Dept. of Computer Science and Engineering, Alagappa University, Karaikudi, Tamilnadu, India
[2]Alagappa University, Karaikudi, Tamilnadu, India

Article Info

Article history:

Received Jul 30, 2018
Revised Sep 14, 2018
Accepted Oct 21, 2018

Keywords:

Cloud computing
Cloud service provider
memory
Resource allocation
virtualization
Resource optimization
CPU utilization

ABSTRACT

Online Nowadays, the enterprises & individuals are contributing their workloads on cloud service providers which are going to increase on daily basis. There are large amount CSP are available to offer virtualized and dynamic resource on pay and use basis. However, there are almost CSP failed to maintain quality of service (QOS) and minimal resource optimization. Some of the existing approaches are highly dedicated on scheduling policy but, it does not considered reliable services with optimized QOS. To offer best solution of above problem, the framework proposes Enhanced Minimal Resource Optimization based Scheduling Algorithm to minimize the resources and maintain the QOS. The method avoids delay in Request-Response model in cloud environment. To avoid overload for resource allocation, the proposed design utilized optimized scheduling policy. Proposed mechanisms utilized optimized service brokering policy to reduce the delay response in cloud environment. The framework also help cloud user to prefer best CSP according to their prior services. The method offers rising trend of resource based structure to reduce the placement churn extensively. Proposed system utilized efficient scheduling policy to transmit data request to CSP with minimal data processing time. The entire utilization is to improve the QOS of cloud service provider in the features of multi-dimensional resource. Based on experimental evaluations, proposed technique improves the CPT (Computation Processing Time) 301.72 milliseconds, BU (Bandwidth Utilization) 20 Mbps, CPUU (CPU Utilization) 5% & MRU (Memory Resource Utilization) 3% on given input parameters compare than existing methodology.

Corresponding Author:

Ravi Mahadevan,
Department of Computer Science and Engineering,
Alagappa University,
Karaikudi, Tamil Nadu.
Email: ravimahadevan.phd@gmail.com.

1. INTRODUCTION

Nowadays, Nowadays, the enterprises and individuals are contributing their workloads to cloud service providers (CSP) have been rapidly increased on daily basis. CSP structure a huge pool of abstracted, virtualized, and dynamically scalable resources for users, pay and use basis. The resources are partitioned into three kinds of services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). IaaS provides storage, CPUs, networks and other low-level resources, PaaS provides programming Graphical User Interface (GUI), and SaaS provides previously created applications.

1.1. Problem

The existing methods worked to optimize the resource allocation in cloud as a task scheduling with QoS constraints. However, the computation time of each subtask is called as a scheduler, which is impractical in virtualized cloud systems. There are large amount CSP are available to offer virtualized and dynamic resource on pay and use basis in cloud. However, there are almost CSP failed to maintain a quality of service (QoS) with minimal resource optimization. The selection of services developed to perform the quality factors becomes more critical and challenging to the achievement of SBSs, particularly when the quality factors are strict. However, existing methods fails to maintain quality-aware service composition to enlarge the success rate of computing. The current methodology concentrated on IaaS where cloud service providers provided dissimilar kinds of resources in the structure of VM occurrences. An IaaS provider provides four kinds of VM occurrences: small (S), medium (M), large (L), and extra large (XL). Software as a Service (SaaS) providers gives a set of applications utilizing the Cloud services provided by an Infrastructure as a Service (IaaS) provider. The technique imagines that the IaaS provider provides a payment only what you utilize strategy on demand and spot virtual machines. A QoS-constrained resource allocation method introduced to submit the user computation task in cloud environment. However, the method is only applicable for single VM instances type.

1.2. Background

Qu et al [1] introduced an uncertain-assessment-aware incentive technique to continually give honest assessments and prefer giving uncertain assessments over untruthful or arbitrary assessments. Grechanik et al [2] designed a method for Provisioning Resources with Experimental Software modelling (PRESTO) to improve cloud elasticity by learning and refining models of under-constrained applications all through the performance testing. Qiu et al [3] discussed a hierarchical correlation model for investigating and evaluating correlated measurements, which included Markov models, queuing theory, and a Bayesian approach. Muelder et al [4] introduced a visually based analysis approach to deal with comprehension and analyzing the performance & behavior of cloud computing frameworks.

Palm et al [5] expressed an ALPINE, a Bayesian framework for cloud performance and prediction. ALPINE depended on Bayesian Networks (BNs) and contained Cloud. Papadopoulos et al [6] developed a PEAS (Performance Evaluation framework for Auto-Scaling) structure for the evaluation of auto-scaling methods. Singh et al [7] described an optimized load balancing framework for the cloud by utilizing Active-Clustering (AC) algorithm and Ant Colony Optimization (ACO) to minimize the complexity and time reduction for a client request for datacenter. Mahdi et al [8] discussed the utilization of adaptive replacement cache (ARC) and probabilistic content placement (PROB) algorithms, which together are known as zone-based- adaptive replacement cache and probabilistic content placement (ZB-ARCPROB). Elmubarak et al [9] enhanced performance based ranking model to help clients for choosing the best services.

Mesbahi et al [10] introduced a performance evaluation, and an analytical correlation between all basic load balancing algorithms & recreated in cloud computing. Gadam et al [11] expressed a combined access probability and data rate as a common metric for cell connection. Han et al [12] developed a traffic load balancing system strive to balance between network utilities, e.g., the average traffic delivery latency, and the green energy usage. In [13] depicted the purpose of the cloud paradigm which enhances the use of network that provided the capabilities of utilizing one node from another node. It described the load balancing between the clients and the servers. In [14] highlighted the tradeoff for offloading. The work provided architecture. The genetic algorithm integrated mobile cloud computing for the purpose of automatic offloading in enhancing the system response time. In [15] explained load balancing plays vital role in cloud performance and its stability. It discussed various load balancing algorithms which helped in distributing the load among the nodes and founded which suited the most.

1.3. Objectives

The paper objectives are following as:

a) To develop a resource allocation framework that can avoid overload in the framework efficiently while minimizing the number of server utilization.
b) To implement the effective Request-Response model for improving the computation process & reduce the traffic of cloud datacenter
c) To design a load prediction algorithm that can capture the future resource usages of applications accurately without looking inside the VMs.
d) To reduce CPU utilization, Bandwidth Utilization & Resource Utilization compare than existing approaches

2. METHOD

The proposed design explains the effective a resource allocation framework for avoiding overload in the framework while minimizing the number of server utilization. The method main objective is to optimize allocated resource, minimize the computation time & minimize resource utilizations. The proposed method describes the system architecture with implementation steps and proposed algorithm details. Figure 1 expresses the workflow of implementation process flow in details. The method improves the QoS of CSP in the features of a multi-dimensional resource. To overcome these problems, Enhanced Minimal Resource Optimization Based Scheduling algorithm is designed for resolving the VMPAC issue considers the presence of resources of multiple categories. It provides automated resource management framework that accomplishes a good stability & reliability of cloud services. In overload avoidance, the capability of a PM should be sufficient to satisfy the resource requires of all VMs running on it. Or else, the PM is overloaded and can lead to degraded performance of its VMs. The proposed technique quantize the quantity of PMs utilization which should be decreased as long as they can still satisfy the needs of all VMs. Idle PMs can be turned off to save computational utilization. The outcomes illustrated that the proposed methodologies evaluate the near-optimal solutions while effectively capturing the dynamic market demand, provisioning the computing resources to match the demand, and generating high revenue. The methodology can capture the increasing development of resource usage patterns and help decrease the placement churn significantly. In addition, the computation processing time of the proposed technique is very less.

2.1. Implementation Pre-processing Steps

2.1.1 Cloud Service Provider

Cloud service provider works a mediator between cloud user and storage server to design effective request-response model in cloud environment. The CSP stores different types of data in a distributed manner on different servers, which geographically current in different places. Cloud service providers deal with enterprise infrastructure, and it offers scalable, protection and consistent service for cloud users with minimal cost.

2.1.2 Cloud User

Cloud user should register as data owner with their basic and credential information details to get the login access. Hence, cloud user can contribute the file or information on deployment server for application users. The uploaded files will be stored in a cloud storage server.

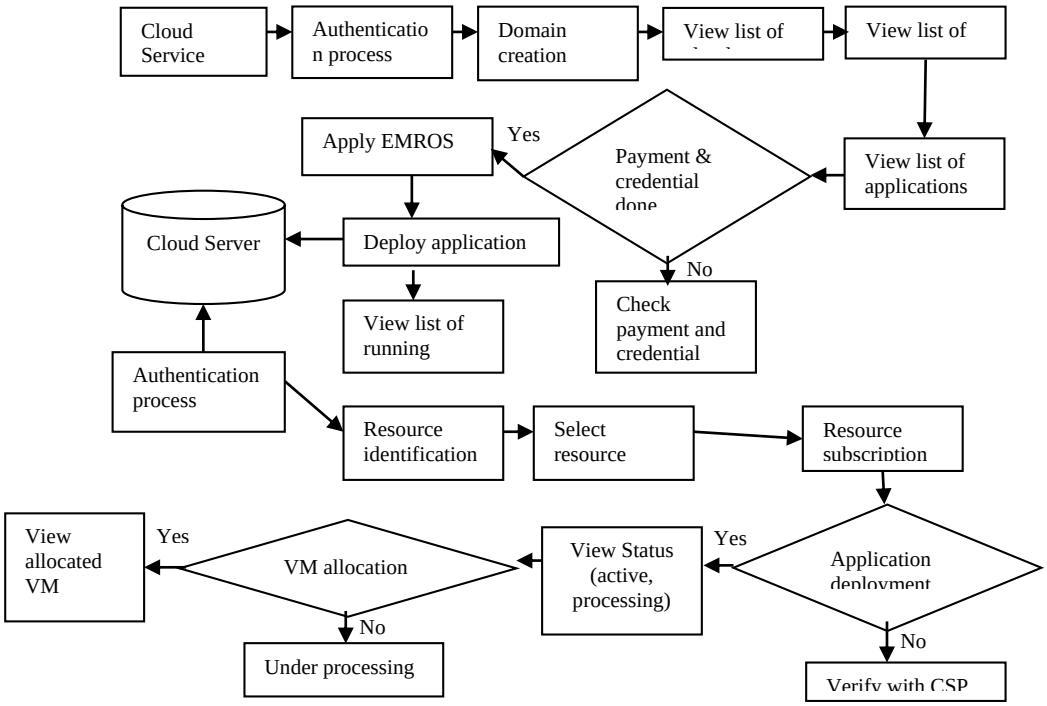

Figure 1. Workflow of Proposed Enhanced Minimal Resource Optimization Based Scheduling Algorithm in Cloud

2.1.3 VM Resource Allocations

The methodology periodically executes, to estimate the resource allocation status based on the calculated upcoming resource requirement of VMs. It specifies the server is overloaded and therefore a few VMs turning on it should be migrated away. The strategy indicates the resource utilization beyond the threshold. It describes a server as if the utilization of the whole of its resources is under a threshold. The framework specifies the server is mostly idle and potential candidate to switch off to save computation power. Though, the methodology does consequently only when the average resource utilization of whole actively utilized servers in the framework are under a computing threshold. A server is actively utilized, and it has at least one VM running, or it is inactive.

2.1.4 VM Mitigation

The methodology is arranged by a list of internet resource in the framework in descending temperature. An objective of the methodology is possible to remove all internet resource or possible keep their temperature as low. For every server, first, choose the VM should be migrated away. If the VM is migrated away, list of VMs arranged by based on the out-coming temperature of the server. The strategy mainly focused on migrating away the most of the VM. For every VM presented in the list, to accommodate the designs observe if methodology can discover a destination server.

2.1.5 Scalability and Flexibility

Scalability and flexibility represented by the performance of data centre. Now, it extends the globally routing through client based on one area data centres to client agent of another area data centres. Proposed methodology calculates internet traffic is routing approximately the world by initiating appropriate broadcast latency and data broadcast delays for distributed computing application.It is additionally serving allotment requests for load balancing rule.

2.2. Enhanced Minimal Resource Optimization Based Scheduling Algorithm

Enhanced Minimal Resource Optimization Based Scheduling Algorithm is designed to represent frameworks composed of thousands of resources, and it creates possible to represent both physical and virtual resources developing particular cloud ideas such as the infrastructure elasticity. The model handles traffic monitoring between user bases and data centres. The default traffic monitoring policy is mapping traffic at nearby data centre regarding network latency from user base. The model represents flow level of application with respective processing workflow and involved factor. Although technique is applicable large number of user bases. The method is capable for servers and data centres; these kinds of systems require specific techniques. The proposed techniques display job scheduling level by level. However, the proposed technique focuses on application to retrieve efficiency of the assigned job to the system from various regions. The proposed technique is handled by the process. The challenge of the technique is to decrease the quantity of active servers through the low load with no sacrificing performance either currently or in the future. In the framework are needed to avoid oscillation. The average consumption of all resources on active servers is invoked by proposed technique. The proposed technique expresses many processes for reducing computation time and energy consumption in computing systems. Its measure various QoS attributes and evaluate the relative ranking of Cloud services.

Here, CSP creates resource information and allocates memory utilization. Cloud users identify the resources based on cost, memory allocation and processing time before finalized the any cloud services. The resource is selected by cloud user for subscription to deploy their application in cloud environments. Cloud user can deploy the application after their credential & payment verification by CSP. Once, deployment process is completed then user can view status of the server, whether, server is active or under processing or inactive. After deployment of cloud user's application, cloud user can view the, computing processing time, CPU utilization, bandwidth utilization & memory utilization. The pseudo code of proposed algorithm is explained below in details:

The Input: Resource allocation, cloud service provider (CSP), cloud user (CU)
Output: Visualize the optimized load & traffic, computing processing time (CPT), CPU utilization (CPUU), bandwidth utilization (BU) & memory utilization (MU)
Procedure:

Start;
Browse Porcess Cloud service provider (CSP) and Cloud user authentication process;
View the available resource with respective CSP details;
Identify the application requirement;
Select the CSP with service utility details;

```
Allocate the resources with memory;
Create userbase;
Assign the task;
 Apply EMROS for Application deployment;
        If Application deployed
                View Status of server (active, processing)
        Else
                View under processing & re-deploy the application
        End If
        If VM allocated
                View Allocated VM
        Else
                Verify with CSP & reallocate VM;
  View list of applications;
  View VMs running status;
```

Display optimized load & traffic, computing processing time (CPT), CPU utilization (CPUU), bandwidth utilization (BU) & memory utilization (MU).

3. RESULTS AND DISCUSSION

3.1. Programming Setup

The proposed method is implemented in Intel i6 Core processor, with 16 GB RAM, 500 GB Memory with Windows7 Ultimate operating systems. The proposed is implemented in NetBeans 8.0, JDK (Java Development Kit) 1.8, MYSQL database 5.5, with Jelastic Cloud server, in Java programming environment. The proposed framework utilizes CloudSim & iText library to deploy and visualize the optimized resource result.

3.2. Input Parameters

The input parameters are explained details in Table 1 to deploy the proposed algorithm to evaluate the efficiency of proposed methodology.

Table 1. Cloud Experimental Evaluation parameter Details

Parameters	Value
Userbase	06
Region	06
Datacenter	4(DC1 and DC2)
Virtual Machine	25(DC_1) 50(DC_1) 75(DC_3) &75 (DC_4)
Data Centre VM	Xen
Number of Process Machine Wise	16
Data Center Processing Speed	100 MIPS
Data Centre VM Policy	Time Shared
Data Centre OS	Windows 7
VM Memory	2048
Data Centre Architecture	X86
Bandwidth	1000 Mbps

3.3. Experimental Result

The proposed method explains the evaluation matrix to calculate the efficiency and utilization of proposed mechanism compare than existing approach. The proposed algorithm is evaluated on different types of input parameters to find out reliability and efficiency. The proposed technique is evaluated with following parameters namely Computation Processing Time, Bandwidth Utilization &, CPU Utilization and Resource Utilization in details.

3.3.1 Computation Processing Time (CPT)

The Computation Processing Time (CPT) computes the time consumption to process the user request from data center in cloud environment. Hence, it processes for retrieve the requested query from database. The computation processing Time is the ratio between the data request and bandwidth consumption per user. The CPT is calculated with mathematical expression in Equation (1).

$$T_{coputprocess} = \frac{D}{BW_{Peruser}} \quad (1)$$

Where Tcomputprocess is total computation time, D is requested data & BWPeruser is total bandwidth utilization user wise

3.3.2 Bandwidth Utilization

The Bandwidth utilization is total network utilization to deploy the application. The bandwidth utilization is expressed in Equation (2) to calculate the request-response model. Bandwidth consumption is a Total allocated bandwidth divided by total number of user requests.

$$BW_{peruser} = \frac{BW_{total}}{N_r} \quad (2)$$

3.3.3 CPU Utilization

CPU utilization expresses the consumption of physical resources for specific task. CPU Utilization is division of Average period of background task (Idle task) without load by average period of background task with load. The CPU utilization is calculated in Equation (3) to evaluate efficiency of proposed methods.

$$CPU\ \text{Utilization} = \frac{\text{Avg Period of Background Task without Load}}{\text{Avg Period of Background Task with Load}} \times 100 \quad (3)$$

3.3.4 Memory Resource Utilization

The memory resource utilization represents the how proposed system is effective to minimize the memory utilization. The memory resource utilization is calculated as the based on the subtraction of allocated memory from buffers and cached memory. The memory resource utilization is calculated mathematically in Equation (4).

$$Memory\ \text{Utilization} = \frac{\text{Allocated}_{\text{Memmory}} - \text{Buffers}_{\text{Memoery}} - \text{Cached}_{\text{Memory}}}{\text{Allocated}_{\text{Memory}}} \times 100 \quad (4)$$

Table 2 displays the Computation Processing Time (CPT), Bandwidth Utilization, CPU Utilization (CPUU) & Memory Resource Utilizations (MRU) for given input parameters. The proposed technique is evaluated on given evaluation parameters with Active Monitoring (AM) [16], Fair Round Robin (FRR) [17], Round Robin (RR) [18] existing approaches. According to Table 2, it noticed that an Enhanced Minimal Resource Optimization Based Scheduling (EMROS) has the best score on every respective constraint for given inputs parameters.

Table 2. Displays the Computation Processing Time (CPT), Bandwidth Utilization, CPU Utilization (CPUU) & Memory Resource Utilizations (MRU)

Learning Algorithms	CPT(ms)	BU (Mbps)	CPUU (%)	MRU (%)
FRR	2049.44	55	19	22
RR	6202.77	50	18	28
AM	6053.51	50	17	17
EMROS	1747.72	30	12	14

According to Figure 2 to 4 performances, it observed that Proposed EMROS shows good result best on CPT, BU, and CPUU & MRU evaluation matrix on given input parameters. In terms of CPT (Computation processing time), FRR (Fair-Round Robin) is closest techniques to Proposed EMROS. However, the FRR fails to optimized the resource and predict the load. Behalf of CPU utilization, Bandwidth Utilization & Memory Resource Utilization (MRU), AM (Active Monitoring) is the closest method. But, it unable to offer efficient Request-Response Model and scheduling policy to minimize the computation processing time and optimized the physical resources. Proposed algorithm maintains equivalent workloads on all the available VMs and the quantity of requests presently assigned to VM. EMROS avoids overload in the framework effectively while minimizing the quantity of servers used. Proposed technique improves the CPT 301.72 milliseconds, BU 20

Mbps, CPUU 5% & MRU 3% on given input parameters compare than existing methodology. Finally, it claims that proposed EMROS methodology is best on all respective constraints

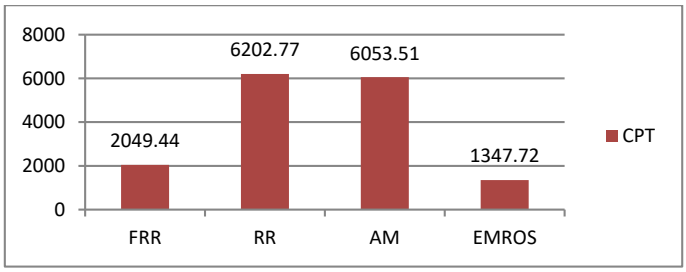

Figure 2. Computation Processing Time (CPT) in milliseconds (ms)

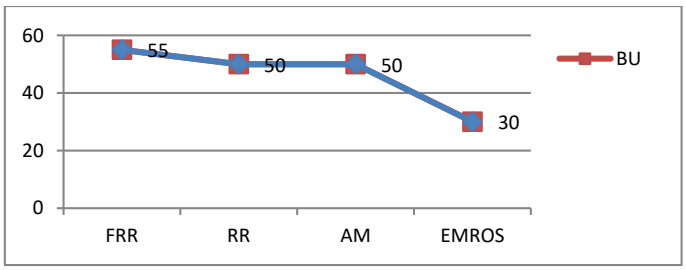

Figure 3. Bandwidth Utilization (Mbps)

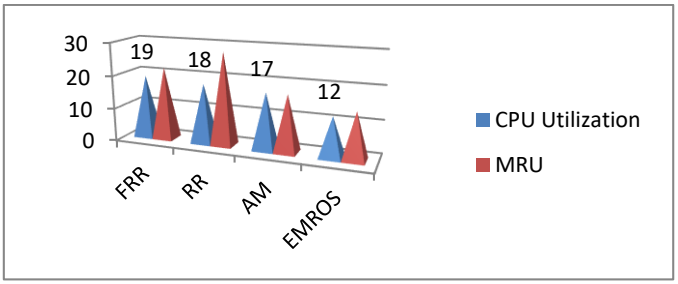

Figure 4. CPU Utilization (BU) and Memory Resource Utilization (MRU) in Percentage

4. CONCLUSION

The paper presents Enhanced Minimal Resource Optimization based Scheduling Algorithm to reducing the resources and maintaining the QoS. The proposed methods avoids overload for resource allocation, & utilized optimize resources. The framework computes the resource utilization based on client requirement. The entire utilization purpose is to enhance the QoS of CSP in the attributes of a multi-dimensional resource. The proposed methodology utilized optimized routing to decrease the traffic in a cloud environment. The framework also helps cloud user to prefer best CSP according to their prior services. The model represents flow level of application with respective processing workflow and involved factor. Although technique is applicable large number of user bases. The method is capable for servers and data centers; these kinds of systems require specific techniques. Its measure various QoS attributes and evaluate the relative ranking of Cloud services. The proposed techniques display job scheduling level by level. However, the proposed technique focuses on application to retrieve efficiency of the assigned job to the system from various regions. Proposed technique improves the CPT 301.72 milliseconds, BU 20 Mbps, CPUU 5% & MRU 3% on given input parameters compare than existing methodology

In future, the paper can be extended to apply the privacy of client application, user log and activity without CSP disclosure. For improving data analytical process, HDFS can be integrated to execute the task effective way in cloud environment.

REFERENCES

[1] Qu, L., Wang, Y., & Orgun, M., "A novel incentive mechanism for truthful performance assessments of cloud services", *In Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems*, pp. 1325-1326, 2016.

[2] Grechanik, M., Luo, Q., Poshyvanyk, D., & Porter, A., "Enhancing rules for cloud resource provisioning via learned software performance models", *In Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering*, pp. 209-214, 2016.

[3] Qiu, X., Dai, Y., Xiang, Y., & Xing, L., "A hierarchical correlation model for evaluating reliability, performance, and power consumption of a cloud service", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 3, pp. 401-412, 2016.

[4] Muelder, C., Zhu, B., Chen, W., Zhang, H., & Ma, K. L., "Visual analysis of cloud computing performance using behavioral lines", *IEEE transactions on visualization and computer graphics*, vol. 22, no. 6, pp. 1694-1704, 2016.

[5] Palm, E., Mitra, K., Saguna, S., & Åhlund, C., "A Bayesian System for Cloud Performance Diagnosis and Prediction", *In Cloud Computing Technology and Science (CloudCom), 2016 IEEE International Conference*, pp. 371-374, 2016.

[6] Papadopoulos, A. V., Ali-Eldin, A., Årzén, K. E., Tordsson, J., & Elmroth, E., "PEAS: A performance evaluation framework for auto-scaling strategies in cloud applications", *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, vol. 1, no. 4, pp. 1-39, 2016.

[7] Singh, R., & Prakash, S., "Enhancement of Resource Allocation using Load Balancing in Cloud Computing", *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, pp. 11-17, 2017.

[8] Mahdi, A. S., & Muniyandih, R. C., "Enhancement of Cloud Performance and Storage Consumption Using Adaptive Replacement Cache and Probabilistic Content Placement Algorithms", *Journal of Theoretical and Applied Information Technology*, vol. 84, no. 3, pp. 376-384, 2016.

[9] Elmubarak, S. A., Yousif, A., & Bashir, M. B., "Performance-based Ranking Model for Cloud SaaS Services", *I.J. Information Technology and Computer Science*, vol. 1, pp. 65-71, 2017

[10] Mesbahi, M. R., Hashemi, M., & Rahmani, A. M., "Performance evaluation and analysis of load balancing algorithms in cloud computing environments", *In Web Research (ICWR), 2016 Second International Conference*, pp.145-151, 2016

[11] Gadam, M. A., Ng, C. K., Nordin, N. K., Sali, A., & Hashim, F., "Hybrid Channel Gain Access Cell Association for Load Balancing in Downlink LTE-Advanced HetNets", *In Computer and Communication Engineering (ICCCE), 2016 International Conference*, pp. 337-342, 2016.

[12] Han, T., & Ansari, N., "A traffic load balancing framework for software-defined radio access networks powered by hybrid energy sources", *IEEE/ACM Transactions on Networking (TON)*, vol. 24, no. 2, pp. 1038-1051, 2016

[13] Talasila, S., Vani, H., Sai, K., Mani, D., Krishna Reddy, V., "Load Balancing Techniques for Efficient Traffic Management in Cloud Environment", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 6, No. 3, pp. 963-973, 2016.

[14] Nadim Akhtar, N. Srinivasan, "Notice of Retraction Intermittently Connected Cloudlet System to Obtain an Optimal Offloading Policy", *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, Vol. 4, No.3, 2016.

[15] Ravi Teja Kanakala, V., Krishna Reddy, V., "Performance Analysis of Load Balancing Techniques in Cloud Computing Environment", *TELKOMNIKA Indonesian Journal of Electrical Engineering*, Vol. 13, No. 3, pp. 568 - 573, 2015.

[16] Awadalla, M. H. A., "Heuristic Approach for Scheduling Dependent Real-Time Tasks", *Bulletin of Electrical Engineering and Informatics*, vol. 4, no. 3, pp. 217-230, 2015.

[17] Jammalamadaka, S. K. R., Duvvuri, K. B. K., Ch, D. A., & Padmini, P., "Building Fault Tollrence within Clouds at Network Level", *International Journal of Electrical and Computer Engineering*, vol. 6, no. 4, pp. 1560-1569, 2016.

[18] Li, J., Meng, X., Wen, J., & Xu, Y., "An improved method of SVM-BPSO feature selection based on cloud model", *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 5, pp. 3979-3986, 2014