Priority based Resource Allocation and Scheduling using Artificial Bee Colony (ABC) Optimization for Cloud Computing Systems

A. Phani Sheetal, K. Ravindranath

***Abstract:** In the Cloud computing systems, the existing works fails to address the failure rate of servers and no mechanism has been provided for fault recovery. In this paper, we propose a Priority based Resource Allocation & Scheduling Technique using Artificial Bee Colony (ABC) Optimization (PRAS-ABC) for cloud environment. Initially, the work load of server and resource requirements of users are predicted by the scout bees by monitoring the past resource utilizations and size of the allocated VM. With this predicted workload, the expected completion time of each server is estimated. Then the tasks requesting the resources are categorized based on the deadline and resource requirements. Then based on the work load and expected completed time, the servers are categorized. Then each category of task will be allocated to the respective category of servers. The proposed approach is implemented in CloudSim environment of Java and compared with existing techniques in terms of resource utilization, percentage of resources successfully allocated, percentage of missed deadlines, average work load of server etc.*

I. INTRODUCTION

A. Cloud Computing

Cloud computing differs from the traditional computing which mainly depends on the personal devices. It allows to share the computing resources from a remote place. It provides flexible and minimum cost access for resources at any place any time. The shared resources can be hardware or software. Cloud offers various services like Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS). [1].

The advantages of Cloud Computing are: (i) Applications can be accessed as utilities throughout the web. (ii) No specific software installation needed for accessing cloud applications. (iii) The PasS service provides various deployment tools and runtime environments (iv) It provides platform independent access to all clients. (v) Supports load balancing [2].

B. Resource Allocation and Scheduling in Cloud

Resource allocation can be static or dynamic. In static allocation, the cloud user requests for fixed amount of resources proactively. But static allocation causes poor or over utilization of resources [3].

In dynamic allocation, the users request for resources depending on the requirements of application by on demand. When the requested resources are not available, the cloud service provider (CSP) allocates from other cloud data centers [4].

Advance resource allocation enables to plan the resources for future such that they can be allocated on demand [5].

C. Problem Identification

In multi agent based VM allocation approach [9], the resource utilization and energy consumption parameters are mainly considered for VM allocation. But it fails to consider the work load prediction or future resource requirements. Moreover, the deadline of each task was not considered.

In [11], when a high priority job (with low deadline) comes in, the low priority job (with high deadline) was preempted allowing the high priority job to run in its resource. But it neither checks the work load of the PMs nor checks the size of the requested resources. In [14], the work load based on future resources requirement is predicted. Then it migrates the VM from a hot spot to a cold spot based on resource utilizations. But this approach did not consider the energy cost for the utilized resources. Moreover, the deadline of each task was not considered.

There are some works available on ABC based task scheduling in cloud. Out of these papers, [22] and [24] are same. They consider the VM load in the fitness function for selecting the VMs. [23] considers makespan time and load balancing (they didnt define these metrics) for fitness function.[25] also considers task completion time and load as fitness function for selecting the VMs. [26] is groups the user requests based on priority. But it considers only make span time as the fitness function for selecting VM.

But the main advantages of our solution over these works are:(i) we have predicted the load and completion time using EWMA from the past values. Only from these predicted values, the fitness function is formed. (ii) Our algorithm does not select individual VM but rather selects a server. (iii) We have classified the servers based on the priority of user requests.

II. RELATED WORKS

Ying Song et al. [6] proposed a two-tiered on-demand resource allocation mechanism, including the local and global resource allocation. A local on demand resource allocation algorithm is applied by the VM with a threshold value. In global resource allocation, the threshold value of local allocation can be adjusted adaptively.

Revised Manuscript Received on April 12, 2019.

A. Phani Sheetal, Research Scholar, Department of CSE, Koneru Lakshmaiah Education Foundation, Guntur, AP, India

K.Ravindranath, Associate Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Guntur, AP, India

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

SivaThejaMaguluriet. al. [7] have discussed about traffic optimal resource allocation algorithms. Sheng Di et. al. [8] have designed a deadline based resource allocation algorithm and an error free technique for job completion time.

Wanyuan Wang et al [9] have introduced a decentralized multiagent (MA) based VM allocation approach. A local negotiation-based VM consolidation mechanism is developed to exchange the assigned VMs of agents for energy cost saving.

GandhaliUpadhye et al [10] have proposed Utility Accrual (UA) approach to associate each task with a Time Utility Function (TUF). The TUF denotes the utility attained by a system at the time when a task is completed to improve the performance.

Chen-Fang Weng et al [12] have designed adaptive neural fuzzy inference system (ANFIS) algorithm for predicting the load and deciding the resource allocation policy for VMS.

GavineKanakadurga et al [13] have presented the many dynamic resource allocation techniques. Resource provisioning was done by parallel processing using different types of scheduling heuristics.

III. PROPOSED SOLUTION

A. Overview

In this paper, we propose a Priority based Resource Allocation &Scheduling Technique using Artificial Bee Colony (ABC) Optimization (PRA-ABC) for cloud environment. Initially, the work load of server and resource requirements of users are predicted by the scout bees by monitoring the past resource utilizations and size of the allocated VM. With this predicted workload, the expected completion time of each server is estimated. Then the tasks requesting the resources are categorized based on the deadline and resource requirements. Then based on the work load and expected completed time, the servers are categorized. Then each category of task will be allocated to the respective category of servers. The proposed approach is implemented in CloudSim environment of Java and compared with existing techniques in terms of resource utilization, percentage of resources successfully allocated, percentage of missed deadlines, average work load of server etc.

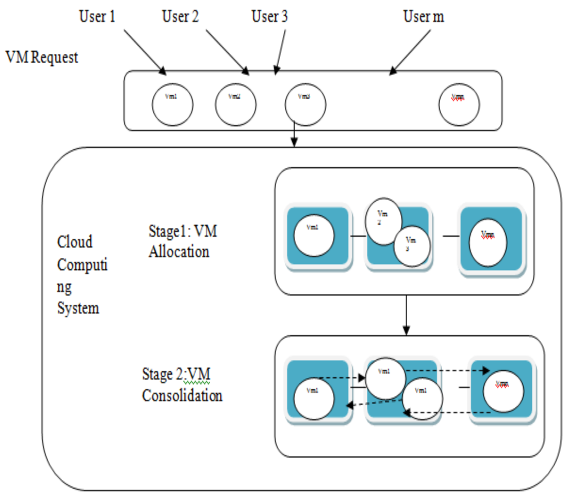

Fig. 1 System architecture

B. Artificial Bee Colony (ABC) Technique

ABC algorithm is a heuristic method of discovering and sharing the food resources of honey bees. Each bee will search for nearby food sources and will share this information to other bees. There are three types of bees involved: Employed, onlooker and scout bees [14][15].

C. Assumptions

Let X = (M, N) be the cloud system

M = {m1, m2, …, m_i} be the set of servers interconnected by the communication network

$\forall(m_i, m_j) \in N$

This indicates that m_i and m_j communicate with each other through only one switch

Let c (p_i, p_j) be the communication distance between servers m_i and m_j .

This is computed as the number of switches along the shortest path between m_i and m_j

Let $\delta = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ be the set of VM resource request required by users.

Let w_i be the amount of resources required by VM $\alpha \in \delta$

Each server owns number of resources that are capable of running multiple VMs.

Let c the amount of resources at the server m_i

D. VM Allocation

A VM allocation $\{\delta(m_1), \delta(m_2), ..., \delta(m_i)$ is defined as mapping of server to set a set of VMs that should satisfy the following two conditions:

1. Each VM should be allocated to at least one server and no VM is allocation to more than one server

$$\cup_{m_i \in m} \Theta(m_i) = \Theta \quad (1)$$

$$\Theta(m_i) \cap \Theta(m_j) = \Phi, \forall 1 \le i, j \le y, i \ne j \quad (2)$$

2. For each server, the total resource requirements of its hosted VM does not exceed its available resources

$$\sum_{\theta j \in \Theta(mi)} c_j \le w_i, \forall 1 \le i \le y \quad (3)$$

E. Prediction of Work Load

Each server maintains a resource utilization history (RUT) per VM which contains the total resources utilized and total time taken to complete each services of users.

The workloads of each VM can be predicted using exponentially weighted moving average (EWMA) of past workloads of all VMs

$$EL(t) = \alpha.EL(t-1) + (1-\alpha).OL(t) \quad (4)$$

where EL and OL are the estimated and observed loads at time t and α is a constant.

Similarly, expected completion time of each server can be determined by estimating EWMA of past completion times of all VMs

$$ECT(t) = \beta.ECT(t-1) + (1-\beta).OCT(t) \quad (5)$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Where ECT and OCT are the expected and observed completion time of VMs at time t and β is a constant.

F. Classification of Users Tasks

When the users submit their service requirements, then the cloud server broker will classify the services and allocates priorities as shown in Table 1

Table. 1 Classification of user tasks

Dead line	Resources size	Priority
Short	High	1
Short	Medium	2
High	High	3
High	Low	4

G. Priority based Resource Allocation & Scheduling Technique

Let Q be the solutions
Let C is the number of optimization parameters
Let T be the maximum number of cycles that the algorithm would run.

1. Initially scout bees are deployed in the network which collects the past resource utilizations and size of the allocated VM.
2. Based on the collected information, work load of server and resource requirements of users are predicted.
3. A fitness function is formed for each server S_i, i=1,2…n using EL and ECT as

$$g_i = (\lambda_1 * EL\ ) + (\lambda_2 * ECT) \quad (6)$$

Where λ_1 *and* λ_2 are the weight values

4. Then roulette wheel selection method (of ABC) will be applied to find the best solution. In this method,

a. The initial positions of food sources are randomly generated.
b. During each iteration, the employed and onlooker bees seek for better solutions by performing neighbor search.
c. For each solution q_i, determine a neighbor b_i using Eq. (7)

$$q_{ij} = b_{ij} + \sigma_{ij}(b_{ij} - b_{kj}) \quad (7)$$

where k $\in$ {1,2,3,…, Q} and k $\neq i$,
σ = random number in the range [-1,1]
j = 1, 2, 3,…, V
k and j are randomly selected

d. A better solution is then selected between b_i and q_i.
e. The onlooker bees are deployed near the food sources using the roulette wheel selection method.
f. It selects a food source at position b_i with a probability Z_i calculated as follows

$$\frac{fit_i}{\sum_{n=1}^{S} fit_n} \quad (8)$$

$$fit_i = \begin{cases} \dfrac{1}{1+g_i}, & if\ g_i \geq 0 \\ 1+abs(g_i) & if\ g_i < 0 \end{cases} \quad (9)$$

where g_i = fitness of the solution

g. A solution is rejected by an employed bee if it could not be enhanced for a fixed number of trials.
h. The employed bee is then changed into a scout bee to generate a new solution randomly.
i. The value of limit is chosen as Q × C.
j. The best solutions are recorded till now.
k. Steps c and h are repeated until T cycles are completed.
l. Determine the global best solution among the best local solutions recorded at each processor.

The scout bees then visit each server and determine the fitness function.

Based on the best solution observed, the servers are categorized as follows:

Table. 2 Category of Servers

Server	Work Load	Expected Completion time
1	Least	Less
2	Medium	Less
3	Less	High
4	High	High

Then each category of task will be allocated to the respective category of servers. The low priority tasks (3) and (4) are preempted when high priority task (1) or (2) arrives.

IV. EXPERIMENTAL RESULTS

The NASA workload [20] has been used as the emulator of Web users requests to the Access Point (AP). This workload represents realistic load deviations over a period time. It comprises 100960 user requests sent to the Web servers during a day. Abnormal deviations in this workload can trigger the load balancing mechanism. Research shows that the pattern of user request arrivals to websites is very much similar to the pattern of this workload. Table 3 shows the experimental parameters assigned in this work.

Table. 3 Experimental Parameters

Parameter	Value
Work load	NADA traces
Resource Utilization Thresholds	$U^{low-thr} = 20\% and U^{high\_thr} = 80\%$
Response Time Thresholds	$RT^{low-thr} = 200ms and RT^{high\_thr} = 1000ms$
Scaling Intervals	$\Delta t = 10min$
Desired Response Time	DRT = 1000ms=1s
Load Balancing Policy	Round-Robin
Configuration of VMs	t2.medium and t2.Large
Maximum On-demand VM Limitation	*Max*VM=10VM
Task and Resources Scheduling Policy	Time-Shared

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

A. Performance Metrics

This section explains experiments conducted to evaluate the performance of the proposed PRAS-ABC scheme in CloudSim Simulator [21]. The performance metrics considered for evaluation are as follows:

Response delay: The response time R(t) represents the latency in the response to user requests. It is calculated (in minutes) according to Eq. (10).

$$R(t)=\frac{\sum_{j=1}^{\mathrm{Re}\,q_{ans}} W_j}{\mathrm{Re}\,q_{ans}} \qquad (10)$$

Where Req_{ans} is the number of answered requests

W_j is the waiting time of each answered requests given by

$W_j = [F_j(t) - A_j(t)] \,.\, Ser_j(t)$ (11)

Where F(t) , A(t) and Ser(t) are the finish time, arrival time and service time. Arrival time and the finish time are the times the request is received from user and the time the response is received by its VM, respectively. Service time is the estimated time required by the incoming user request.

Throughput: It represents the ratio of answered requests (Req_{ans}) to the total number of received requests (Req_{rec}) in percentage.

Resource utilization: It represents the mea% of the CPU utilization by calculating the average of all VMs utilization.

$$R_u=\frac{\sum_{j=1}^{RVM} U(VM_j)}{RVM} \qquad (12)$$

Here RVM is the number of rented VMs and $U(VM_j)$ represent the utilization of VM_j

Missed deadlines (M_d): Requests experiencing the response delay of more than the requested deadline (R_d) are considered as missed deadline.

$$M_d = \sum_{j=1}^{Tot\,\mathrm{Re}\,q} W_j - R_d \qquad (13)$$

B. Results

Performance of PRAS-ABC

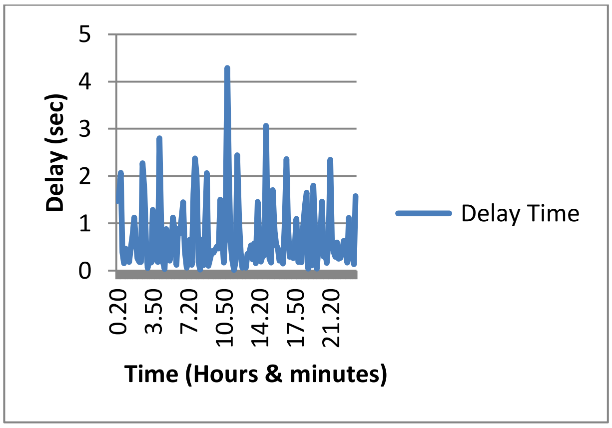

Fig. 2 Time Vs Response Delay

In this experiment time taken as x-axis and the corresponding delay is calculated. The performance can be evaluated through the graph.

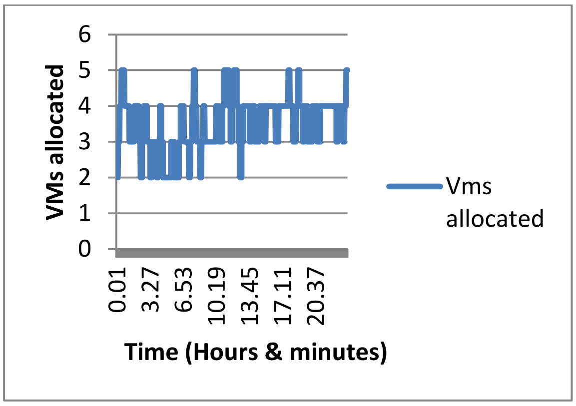

Fig. 4 Time Vs VMs Allocated

In this experiment time taken as x-axis and the corresponding VMs allocation is calculated. The performance can be evaluated through the graph.

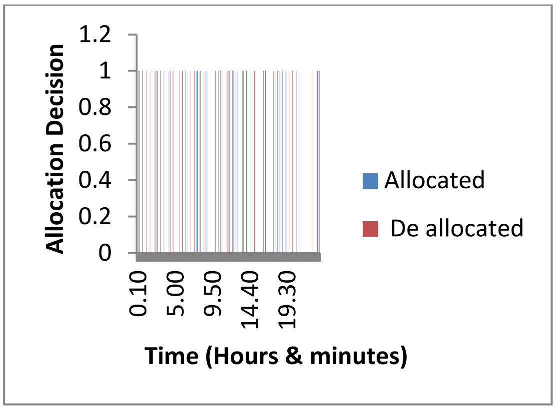

Fig. 5 Time Vs Allocation Decision

In this experiment time taken as x-axis and the corresponding allocation decision is calculated. The performance can be evaluated through the graph.

C. Performance of PRAS-ABC with Suprex

In this section, the performance of the proposed PRAS-ABC is compared with the super professional executor (Suprex) [19]. In Suprex, if the resources are under-provisioned, the executor adds a new VM. On the other hand, if the resources are over-provisioned, the executor releases a VM by selecting a VM from on-demand.

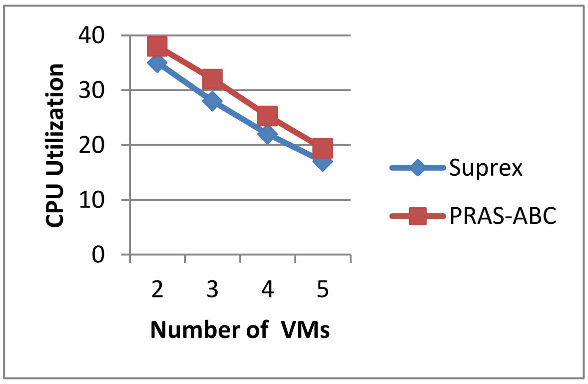

Fig. 6 No. of VMs Vs CPU Utilization

Figure 6 shows the CPU Utilization measured for PRAS-ABC and Suprex when number of VMs are varied.

The VMs are increased from 2 to 5, as we can see from the figure, the CPU Utilization of PRAS-ABC decreases from 38 to 19.4 , the CPU Utilization of Suprex decreases from 35 to 17. Hence the CPU Utilization of PRAS-ABC is 13% of higher when compared to Suprex.

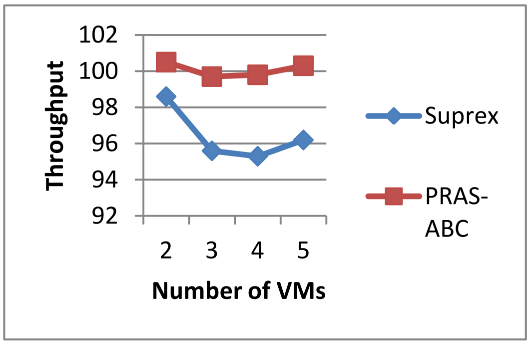

Fig. 7 No. of VMs Vs Throughput

Figure 7 shows the Throughput measured for PRAS-ABC and Suprex when number of VMs are varied. The VMs are increased from 2 to 5, as we can see from the figure, the Throughput of PRAS-ABC decreases from 100.5 to 100.3, the Throughput of Suprex decreases from 98.6 to 96.2. Hence the Throughput of PRAS-ABC is 4% of higher when compared to Suprex.

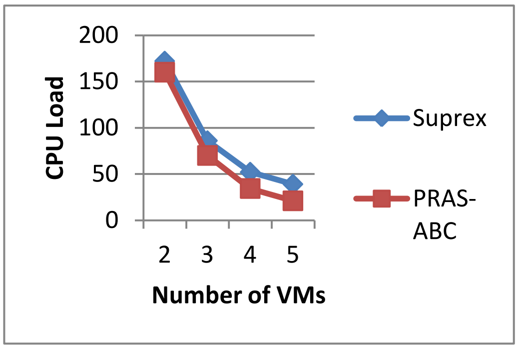

Fig. 8 No. of VMs Vs CPU Load

Figure 8 shows the CPU Load measured for PRAS-ABC and Suprex when number of VMs are varied. The VMs are increased from 2 to 5, as we can see from the figure, the CPU Load of PRAS-ABC decreases from 160 to 21, the CPU Load of Suprex decreases from 172 to 39. Hence the CPU Load of PRAS-ABC is 42% of higher when compared to Suprex.

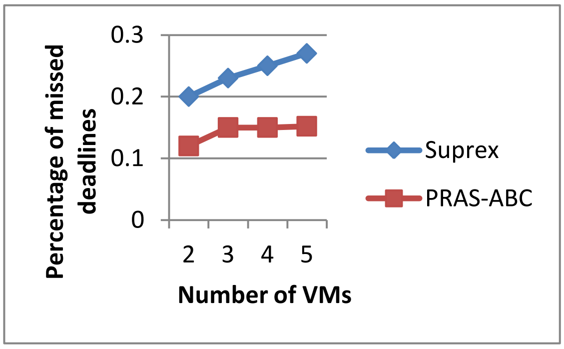

Figure 9: No. of VMs Vs Percentage of missed deadlines

Figure 9 shows the percentage of missed deadlines for the requested services when the number of VMs is varied. The figure shows that PRAS-ABC attains 66% lesser missed deadlines than Suprex scheme.

The average values of each metrics for both the schemes are shown in Table 4.

Table. 4 Average Values for both the schemes

Scheme	Time (Hrs & Mins)	CPU Utili-zation	Throughput	Delay (sec)	% of Missed deadline	CPU Load
PRAS-ABC	0.01-23.01	26.52	100.97	0.7218	0.1695	59.404
Suprex	0.01-23.01	23.52	96.2	0.788	0.2095	71.404

V. CONCLUSION

In this paper, we have proposed a Priority based Resource Allocation & Scheduling Technique using Artificial Bee Colony (ABC) Optimization (PRA-ABC) for cloud environment. Initially, the work load of server and resource requirements of users are predicted by the scout bees by monitoring the past resource utilizations and size of the allocated VM. With this predicted workload, the expected completion time of each server is estimated. Then the tasks requesting the resources are categorized based on the deadline and resource requirements. Then based on the work load and expected completed time, the servers are categorized. Then each category of task will be allocated to the respective category of servers. The proposed approach have been implemented in CloudSim environment of Java and compared with existing techniques in terms of resource utilization, percentage of resources successfully allocated, percentage of missed deadlines, average work load of server etc.

REFERENCES

1. 1.Kamini Bharti and Kamaljit Kaur, "A Survey of Resource Allocation Techniques in Cloud Computing", International Journal of Advanced Computer Engineering and Communication Technology (IJACECT), ISSN (Print): 2319-2526, Volume-3, Issue-2, 2014.
2. 2.Bhavani B H and H S Guruprasad, "Resource Provisioning Techniques in Cloud Computing Environment: A Survey", International Journal of Research in Computer and Communication Technology, Vol 3, Issue 3, March- 2014.
3. 3.PankajSareen,Parveen Kumar and Tripat Deep Singh, "RESOURCE ALLOCATION STRATEGIES IN CLOUD COMPUTING", International Journal of Computer Science & Communication Networks,Vol 5(6),358-365.
4. 4.N. Asha and G. Raghavendra Rao, "A Review on Various Resource Allocation Strategies in Cloud Computing", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 7, July 2013.
5. 5.Ronak Patel and Sanjay Patel, "Survey on Resource Allocation Strategies in Cloud Computing", International Journal of Engineering Research & Technology (IJERT),Vol. 2 Issue 2, February- 2013.

6. 6.Ying Song, Yuzhong Sun and Weisong Shi, “A Two-Tiered On-Demand Resource Allocation Mechanism for VM Based Data Centres”, IEEE transactions on services computing, ISSN: 1939-1374, vol. 6, no. 1, Jan. 2013.
7. 7.Siva ThejaMaguluri, R. Srikant, Lei Ying,”Heavy Traffic Optimal Resource Allocation Algorithms for Cloud Computing Clusters”, 24th International Teletraffic Congress, 2012, Article No. 25.
8. 8.Sheng Di, Cho-Li Wang, “Error-Tolerant Resource Allocation and Payment Minimization for Cloud System”, IEEE Transactions On Parallel And Distributed Systems, Vol. 24, No. 6, June 2013, pp 1097-1106, DOI 10.1109/TPDS.2012.309.
9. 9.Wanyuan Wang, Yichuan Jiang and Weiwei Wu,"Multiagent-Based Resource Allocation for Energy Minimization in Cloud Computing Systems", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS,2016.
10. 10.GandhaliUpadhye and TruptiDange, "Cloud Resource Allocation as Non-Preemptive Approach",2nd International Conference on Current Trends in Engineering and Technology, ICCTET’14,2014.
11. 11.Saraswathi AT , Kalaashri.Y.RA and S.Padmavathi c, "Dynamic Resource Allocation Scheme in Cloud Computing", Elsevier, Procedia Computer Science 47 (2015) 30 – 36,2015.
12. 12.Chen-Fang Weng and Kuo-Chen Wang,"Dynamic Resource Allocation for MMOGs in Cloud Computing Environments", IEEE,2012.
13. 13.GavineKanakadurgaSantanB.Veeramallu, "Dynamically Allocating the Resources Using Virtual Machines",(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 4646-4648.
14. 14.Zhen Xiao,Weijia Song, and Qi Chen, "Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment", IEEE TRANSACTION ON PARALLEL AND DISTRIBUTED SYSTEMS(TPDS),2012.
15. Yeongho Choi and Yujin Lim,” Optimization Approach for Resource Allocation on Cloud Computing for IoT”, International Journal of Distributed Sensor Networks, Hindawi, Volume 2016, Article ID 3479247, 6 pages
16. 16.Ajit Kumar, Dharmender Kumar, S. K. Jarial,” A Review on Artificial Bee Colony Algorithms and Their Applications to Data Clustering”, Cybernetics And Information Technologies, Volume 17, No 3, 2017
17. 17 S. Mingprasert and R. Masuchun,” Adaptive Artificial Bee Colony Algorithm for solving the Capacitated Vehicle Routing Problem”, IEEE 2017
18. Punit Gupta and S. P. Ghrera,” Power and Fault Aware Reliable Resource Allocation for Cloud Infrastructure”, International Conference on Information Security & Privacy (ICISP2015), Elsevier, December 2015, Nagpur, INDIA
19. Mohammad SadeghAslanpour, Mostafa Ghobaei-Arani, Adel NadjaranToosi,” Auto-scaling Web Applications in Clouds: A Cost-Aware Approach”, Journal of Network and Computer Applications, 17 July 2017
20. M. F. Arlitt and C. L. Williamson, "Internet web servers: Workload characterization and performance implications," IEEE/ACM Transactions on Networking (ToN), vol. 5, pp. 631-645, 1997.
21. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Rose and RajkumarBuyya,” CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms”, SOFTWARE – PRACTICE AND EXPERIENCE, 2011
22. B. Kruekaew and W. Kimpan,” Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony:”
23. 23.NASRIN HESABIAN and HAMID HAJ SEYYED JAVADI, “Optimal Scheduling In Cloud Computing Environment Using the Bee Algorithm”, International Journal of Computer Networks and Communications Security, VOL. 3, NO. 6, JUNE 2015, 253–258
24. B. Kruekaew and W. Kimpan,” Heuristic Task Scheduling with Artificial Bee Colony Algorithm for Virtual Machines”, IEEE Joint 8th International Conference on Soft Computing and Intelligent Systems and International Symposium on Advanced Intelligent Systems, 2016
25. FatemehRastkhadiv and Kamran Zamanifar,” Task Scheduling Based On Load Balancing Using Artificial Bee Colony In Cloud Computing Environment”, International Journal of Advanced Biotechnology and Research (IJBR), Vol-7, Special Issue-Number5-July, 2016, pp1058-1069