Dynamic Virtual Machine Allocation Policy for Load Balancing using Principal Component Analysis and Clustering Technique in Cloud Computing

Law Siew Xue, Nazatul Aini Abd Majid and Elankovan A. Sundararajan
Faculty Information Science and Technology,
Universiti Kebangsaan Malaysia (UKM), Malaysia.
nazatulaini@ukm.edu.my

***Abstract*—The scalability and agility characteristics of cloud computing allow load balancing to reroute workload requests easily and to enhance overall accessibility. One of the most important services for cloud computing is Infrastructure as a Service (IaaS). There is a large number of physical hosts in a cloud data center for IaaS and it is quite difficult to arrange the allocation of the workload requests manually. Therefore, different load balancing methods have been proposed by researchers to avoid overloaded physical hosts in the cloud data center. However, fewer works have used multivariate analysis in cloud computing environment for considering the dynamic changes of the computing resources. Thus, this work suggests a new Virtual Machine (VM) allocation policy for load balancing by using a multivariate technique, Principal Component Analysis (PCA), and clustering technique. Moreover, PCA and clustering techniques were simulated on a cloud computing simulator, CloudSim. In the proposed allocation policy, a group of VMs were dynamically allocated to physical hosts. The allocation was based on the clusters of hosts according to their similar features in computing resources. The clusters were formed using PCA and a clustering technique based on variables related to the physical hosts such as Million Instructions Per Second (MIPS), Random Access Memory (RAM), bandwidth and storage. The results show that the completion time for all tasks has decreased, and the resource utilization has increased. This will optimize the performance of cloud data centers by effectively utilizing the available resources.**

***Index Terms*—Virtual Machines; Allocation Policy; PCA Technique; Clustering; Cloud Computing; CloudSim.**

I. Introduction

Cloud computing is not only about providing computing services via virtualized computing resources but also emphasizing that the services can be used everywhere. According to the National Institute of Standards and Technology (NIST), the definition of cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1].

Service providers allow clients to specify their resource requirements in terms of storage, CPU cores, memory and networking capabilities [2] to support cloud scalability and flexibility. Cloud computing uses virtualization technology to achieve the objective of providing computing resources as a utility [3]. The services offered by cloud computing can be divided into three main categories, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

In order to provide these services, the infrastructure of cloud computing is shared and made with a large degree of data redundancy. The requirements are fulfilled in the cloud computing environment by creating a VM [4, 5]. The computing resources are then allocated to the VM based on the requirements requested from the cloud applications. Hundreds of thousands of physical servers are hosted in a cloud data center. These physical hosts continuously need to process huge amounts of data [6]. This leads to difficulty in terms of allocation arrangement for the workload requests manually [7].

A number of load balancing methods have been proposed by researchers to avoid overloaded physical hosts in the cloud data center. Load balancing is a critical part of the cloud computing lifecycle because load balancing is needed to manage and deal with a lot of loads dynamically in cloud computing environment [4].

However, most researchers do not concentrate on variables or Component Analysis in heterogeneous cloud computing environment as in [8-15]. PCA can handle heterogeneous sets of variables. Therefore, this work suggests using PCA and clustering technique for VM allocation policy in cloud computing environment. This work provides dynamic load balancing policy based on different hosts and VM variables in a cloud environment like memory, speed, storage, and bandwidth.

The remaining paper is organized as follows: Section II presents related work. The CloudSim allocation policy for simulation is discussed in Section III. The proposed PCA and clustering model is presented in Section IV. Section V covers the result and discussions. This paper is concluded in Section VI.

II. Related Work

A VM is an emulation of a computer system based on computer architectures, while simultaneously providing the functionality of a physical computer [16]. VM normally contains virtual CPU cores, the required (CPU capacity per core), RAM and disk sizes. Additionally, bandwidth and latency can also be the requirements needed by the clients for their intended VM. The VM's resource requirements can differ according to time. It can be stable or continually changing depending on the type of application processed in the VM.

The rented VMs features include speed (MIPS), amount of RAM storage, and network bandwidth. For this situation, management entities are important to deal with VM allocation. The management entities should follow the client demands for different VM types and allocate these requests to physical machines depends on the policies defined by the Cloud Provider [6]. VM allocation is a term used in cloud computing for virtual distribution of physical machines between the datacenters [17]. It provides a way to allocate VM to a specific datacenter. Different policies make this allocation efficient and easy to understand.

Many works have been done on resource scheduling and allocation in cloud computing. As a result, a lot of new algorithms, management techniques and different methods for resource scheduling in cloud computing are formed. A series of static, dynamic and hybrid task scheduling methods in cloud computing have been proposed by researchers in Table 1.

Table 1
Comparative analysis of related studies of load balancing in cloud computing

Research	Method	Static/ Dynamic	Makespan	Resource utilization
[8]	Round Robin	Static	No	Yes
[9]	Short Job Scheduling	Static	No	Yes
[10]	Genetic Algorithm	Static	No	No
[11]	Max Min and Min Min	Dynamic	Yes	Yes
[12]	Honeybee behavior	Dynamic	Yes	Yes
[13]	Stochastic Hill Climbing	Dynamic	No	Yes
[14]	Throttled	Dynamic	No	Yes
[15]	Bayesian and clustering	Dynamic	Yes	Yes

Static algorithms are suitable for homogeneous and steamy environments. The main drawback for static algorithms like Round robin [13], Short Job Scheduling [9], Max-Min and Min Algorithm [11] is the node selection for a process allocation is made at the time of creation of process and it cannot be changed during the execution of a process. This may lead to a node overload sometimes and results in poor performance of the overall system [18]. Service composition system in cloud computing should be designed dynamically to overcome intrinsic changes in cloud environments [19].

Therefore, dynamic load balancing methods have been the solutions to overcome intrinsic changes in cloud environments. Most of the dynamic methods used statistical techniques and Artificial Intelligence (AI) such as Throttled Load Balancing [20], Honeybee behavior Load Balancing [12], Genetic Algorithm load balancing [10] and Stochastic Hill Climbing load balancing [21]. However, most of them focus on short-term and not batch processing, and this leads to a long waiting time to complete every task request.

Although [15] proposed Bayesian and clustering load balancing for batch processing, the heterogeneous in cloud environment should also be emphasized in the load balancer. Many cloud applications largely assume a homogeneous environment. To take full advantage of the available hardware(s), cloud-oriented applications must be heterogeneous-aware [22]. Therefore, a new load balancing that supports dynamic, batch-processing and heterogeneous technique is needed to develop a load balancing in cloud computing.

One of the viable approaches is to use a multivariate technique for load balancing in cloud computing. PCA can be generalized as correspondence analysis (CA) in order to handle qualitative variables and as multiple factor analysis (MFA) in order to handle heterogeneous and large data sets of variables [23, 24]. A clustering technique and PCA technologies for modeling and analyzing heterogeneous dataset using binary coded factorial analysis where whereas proposed by [25]. The results of the research show that PCA and clustering have great potential in extracting scalable knowledge from the heterogeneous dataset. Therefore, the objective of this study is to develop a new VM allocation policy using PCA and clustering technique that incorporates the dynamic changes in computing resources to allocate the VM to the physical hosts dynamically.

III. EXPERIMENTAL SETUP USING CLOUDSIM

CloudSim is an extensible simulation framework that allows seamless modeling, simulation, and experimentation of emerging cloud computing infrastructures and application services [26]. In implementing CloudSim for the proposed VM allocation policy in this research, the main steps are:

1. Download and Install Eclipse DSL Tools Version: Mars.2 Release (4.5.2)
2. Download a Java runtime environment which is Java JDK 1.8 because Java is the base platform for eclipse.
3. Download CloudSim package from CloudSim GitHub GitHub.
4. Extract the CloudSim package.
5. Import the CloudSim package into the workspace of the eclipse. The CloudSim package contains source files, jar files, classes and some examples to understand the behavior of cloud computing simulation.
6. Download commons-math3-3.4.1-bin.zip from Apache common math website CloudSim using math function from Apache Math.
7. Configure and run the CloudSim package.

CloudSim supports VM scheduling at two levels which are host level and VM level to enable simulation of different policies for different levels of performance separation. At the host level, the overall computing power for each core in a host will be assigned to each VM. At the VM level, the specific amount of the available processing power by the VMs is allocated to every cloudlet or task. CloudSim implements the time-shared and space-shared resource allocation policies at each level. In implementing the proposed VM allocation policy, the main step is class creation. It is necessary to know the location where modification can be made. VMallocationPolicy.java class is the place where the allocation of VM at the host level is implemented or extended. Moreover, the allocateHostForVm (VM, hs) function for allocating a physical to a VM is finally revoked.

A scenario was simulated on CloudSim using Eclipse where Cloudlet Scheduler Space Shared was used for scheduling VM layer and VM Scheduler Time Shared was used for scheduling host layer. A datacenter with 20 physical hosts was created and these physical hosts had different available computing resources (Host MIPS, RAM, storage, and bandwidth) as listed in Table 2.

Table 2
Variable for Hosts

Host	Host MIPS	Host RAM	Host Storage	Host Bandwidth (Mbps)
0	1023	512	5000	5000
1	2048	1800	8000	8000
2	250	124	1000	1000
3	2048	1600	7000	7000
4	2600	1240	6000	6000
5	2500	1530	5500	5800
6	3300	2500	6000	6000
7	1200	980	6000	6000
8	2272	1792	8482	7392
9	3288	2048	9500	9215
10	780	850	1300	1200
11	2000	1300	6000	7500
12	2900	1850	3500	8000
13	1738	1524	5781	9200
14	1900	1358	7200	4402
15	600	2000	2400	3000
16	2500	1000	2679	2815
17	1312	1024	8000	2250
18	900	952	3142	8473
19	1000	2048	4336	4704

30 VMs were created with different variable values as listed in Table 3.

Table 3
Variable for VM

VMID	VM MIPS	VM RAM	VM Size	VM Bandwidth (Mbps)
0	457	363	3000	3000
1	566	149	1900	1800
2	820	711	3200	2500
3	715	609	2800	3023
4	463	478	1989	2380
5	235	121	870	939
6	2038	1574	6842	6931
7	2528	1092	5906	5973
8	1657	975	2931	3186
9	842	592	2471	2769
10	1500	1620	3488	2926
11	1759	880	2509	3072
12	512	330	1800	2484
13	412	283	2658	2142
14	268	357	1439	1368
15	1421	1024	4500	3229
16	617	512	2183	1932
17	234	256	1700	2125
18	2125	1920	9422	8136
19	753	842	1256	1147
20	1916	1274	5982	4435
21	2857	1833	3344	4974
22	1392	1130	3687	4625
23	340	233	2090	2567
24	1862	1321	7176	3389
25	596	1830	2380	2947
26	2412	932	2500	2423
27	1277	1024	7837	2238
28	833	912	3026	3246
29	984	1850	3912	2423

IV. VM Allocation Policy Based on PCA and Clustering

A dynamic VM allocation policy based on PCA and clustering was proposed to make an analysis of the variables in cloud computing environment. PCA is a data reduction approach that is able to extract most of the important data from a huge multivariable process onto a reduced dimensional PCA model (e.g. [27]) A PCA model is typically built from a few principal components.

Since every physical host contains variables (multivariable) such as million instructions per second (MIPS), random access memory (RAM), bandwidth and storage, PCA is used to reduce the dimension of variables for hosts without losing too much of the host's information. Therefore, the scores produced from PCA can be used later in clustering. Clustering technique will group objects based on the information found in the data describing the object or their relationships [28].

This study proposes using K-means clustering technique to determine the main groups in a set of hosts. The greater the similarity within a group or the greater the difference between groups, the more distinct will the clusters be. The proposed policy is to extract the current computing resources of the physical hosts and to cluster the hosts based on their similar features. A new coming requested task will be assigned to a selected VM. The computing resources of the VM will then be extracted and matched with the hosts' clusters to select the most suitable physical hosts for deploying the requested tasks. The PCA and clustering algorithm consists of two main phases.

A. First phase: VM allocation model and clusters based on PCA and clustering

PCA technique was used to extract main features of the physical hosts to form a reference model for future VM allocation. The main steps in this first phase are:

1. Subtract the mean for a training data set: subtract the mean from each of the data dimensions which is host MIPS, RAM, Bandwidth, and Storage. The mean subtracted is the average across each dimension. This produces a data set with zero means.
2. Perform PCA to the covariance matrix
3. Calculate the eigenvectors and eigenvalues
4. Construct feature vectors
5. Derive scores for the training dataset
6. Perform K-means clustering.
7. Selects K centroids (K rows chosen at random) from the training dataset.
8. Assign each data point from training data set to its closest centroid.
9. Identify the relationship between clusters and the capacity of the computing resources.

B. Second phase: Dynamic VM allocation

This phase is to identify and allocate the appropriate group of hosts that match with the new arrivals of VM. The main steps in this second phase are:

1. Calculate scores for a new coming VM (containing the VM parameters MIPS, RAM, size, and bandwidth) by using feature vectors from the first phase.
2. Measure the VM score with the centroid of clusters using Euclidean distance and find the nearest cluster (the most suitable cluster of hosts based on the capacity of the computing resources).
3. The new VM will be dynamically allocated to the nearest physical host that matches with the existing capacity.

V. RESULTS AND DISCUSSION

In the first phase, PCA was performed to a covariance matrix for a training data set as listed in Table 1. The eigenvalues for the training data set for the first Principal Component (PC1) until the fourth Principal Component (PC4) are listed in Table 4. The variances of the components extracted using PCA in CloudSim are shown in Figure 1.

Table 4
Variable for Hosts

PC	Eigenvalues
PC1	10016816.839
PC2	2704918.242
PC3	535504.411
PC4	179837.282

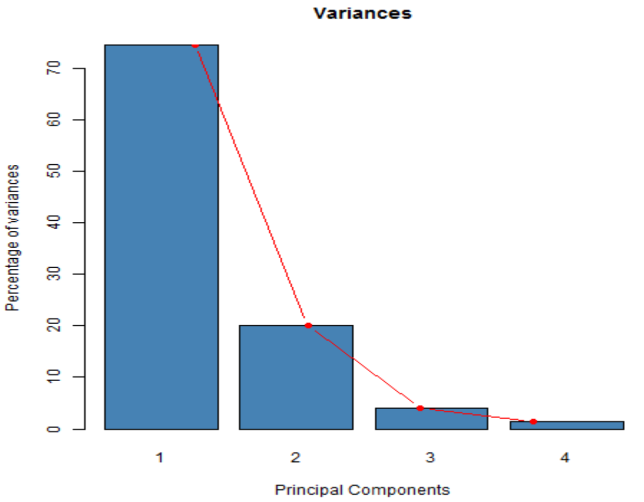

Figure 1: The variances of the components extracted using PCA in CloudSim

The K-Means clustering method was used for the next step. After clustering, the result shows that three clusters of hosts have been created as shown in Figure 2. The first cluster in red was having the medium available computing resources. The second cluster in green is defined as the cluster which has the largest available computing resources. The third cluster in blue has the smallest value of available computing resources. The centroids for these clusters are listed in Table 5.

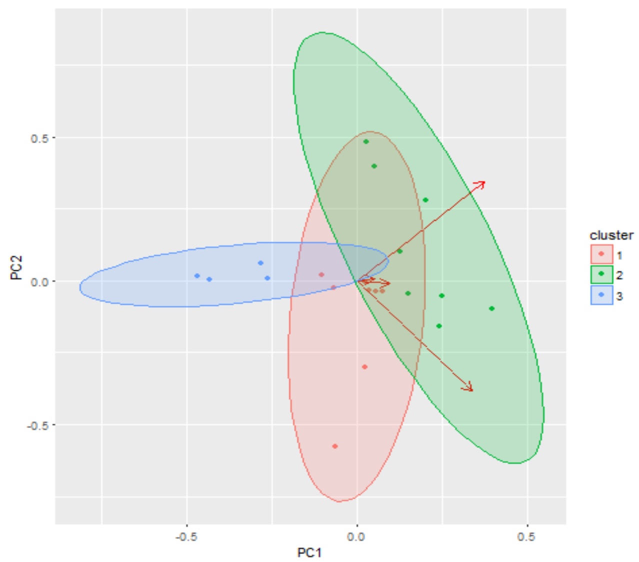

Figure 2: Clustering result using training dataset

Table 5
Cluster centroids for each cluster

Cluster centroids	Centroids coordinate
Cluster 1	-211.03, -32.22
Cluster 2	625.53, 2369.68
Cluster 3	-1511.09, 162.93

The second phase of the policy was started after creating the three clusters. The distance of each VM was calculated from the three cluster centroids using Euclidean Distance to find a cluster.

Then the VM is allocated to the first host ID in the shortest distance from the centroid of the cluster. After all the VMs had been allocated using the VM allocation policy based on PCA and clustering, the results from the CloudSim console are shown in Table 6.

A. A completion time (Makespan)

The completion time for all cloudlets or tasks as shown in Table 5 is 175.15 milliseconds using the proposed PCA and clustering allocation policy. 25 VMs were successfully created out of 30 VMs in this policy and had run cloudlets simultaneously. This allocation policy has reduced the Makespan compared to the default policy (FCFS) in CloudSim where the completion time for all cloudlets 340.52 milliseconds is because only 22 VMs out of 30VMs had been created successfully. The results show that the proposed PCA and clustering allocation policy has improved the Makespan for the VM allocation Policy. This is because the proposed strategy uses the clusters to allocate the new task. This will reduce the time needed to search for a physical host that is suitable with the new task because the search has been narrowed down to a particular cluster that has similar computing capacity.

B. Resource utilization

The optimal use of resources can prevent excessive load in certain physical hosts and wastages in the physical host resources in cloud computing. The calculation of average resource utilization is shown in Equation (1).

$$Average\ CPU\ Utilization\ = \frac{MIPS - availableMIPS}{MIPS} \times 100 \qquad (1)$$

CPU resource utilization for every host had been calculated and recorded. Figure 3 shows that there are five physical hosts with zero resource utilization which are physical hosts with HostID 2, 7, 10, 15, 18 in RoundRobin allocation policy. Only one physical host has resource utilization of more than 80 percent in RoundRobin policy.

For FCFS allocation policy shown in Figure 4, every physical host resource has been utilized. There are six physical hosts with resource utilization of more than 80 percent. This shows that the FCFS allocation policy has optimal use of resources compared to the RoundRobin allocation policy.

CPU resource utilization of each physical host for the proposed PCA and clustering allocation policy is shown in Figure 5. The CPU resource for a physical host with HostID 9 has not been used. However, there are 13 physical hosts having utilization resource of more than 80 percent.

Table 6
Results from the CloudSim console using PCA and clustering policy

ID	Status	Datacenter ID	VM ID	Time	Start Time	Finish Time
7	Success	2	7	15.82	0.1	15.92
6	Success	2	6	19.63	0.1	19.72
21	Success	2	24	21.48	0.1	21.58
11	Success	2	11	22.74	0.1	22.84
8	Success	2	8	24.14	0.1	24.24
10	Success	2	10	26.67	0.1	26.77
15	Success	2	15	28.15	0.1	28.25
19	Success	2	22	28.74	0.1	28.84
23	Success	2	27	31.32	0.1	31.42
9	Success	2	9	47.45	0.1	47.55
24	Success	2	28	48.02	0.1	48.12
2	Success	2	2	48.78	0.1	48.88
18	Success	2	19	53.12	0.1	53.22
3	Success	2	3	55.94	0.1	56.04
16	Success	2	16	64.83	0.1	64.93
22	Success	2	25	67.11	0.1	67.21
1	Success	2	1	70.67	0.1	70.77
12	Success	2	12	78.12	0.1	78.22
4	Success	2	4	86.39	0.1	86.49
0	Success	2	0	87.53	0.1	87.63
13	Success	2	13	97.09	0.1	97.19
27	Success	2	2	48.78	48.88	97.66
28	Success	2	3	55.94	56.04	111.99
20	Success	2	23	117.65	0.1	117.75
26	Success	2	1	70.67	70.77	141.44
14	Success	2	14	149.25	0.1	149.35
5	Success	2	5	170.21	0.1	170.31
17	Success	2	17	170.94	0.1	171.04
29	Success	2	4	86.39	86.49	172.88
25	Success	2	0	87.53	87.63	175.15

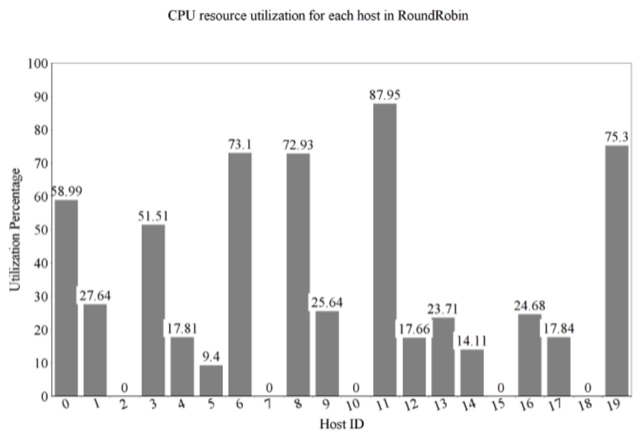

Figure 3: CPU resource utilization for RoundRobin

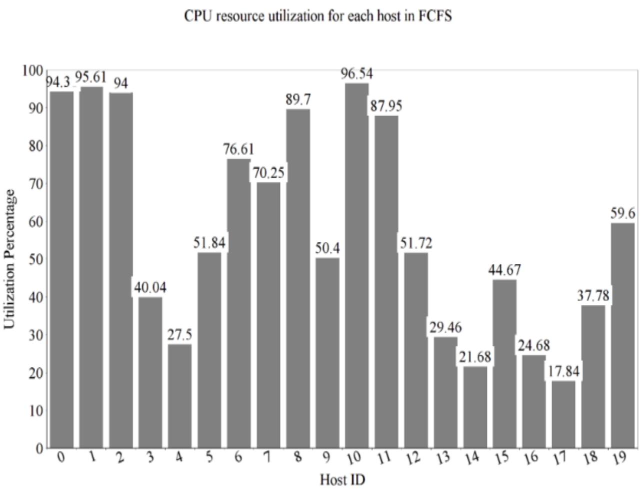

Figure 4: CPU resource utilization for FCFS

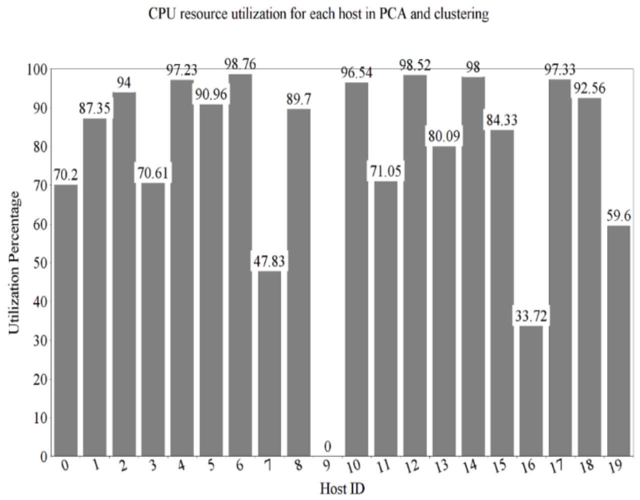

Figure 5: CPU resource utilization for PCA and clustering

The results of the comparison for Round Robin (Figure 3), FCFS (Figure 4) and the proposed PCA and clustering allocation model (Figure 5) show a better performance for load balancing for the proposed model in terms of resource utilization. This is because the proposed load balancing strategy considered more variation for physical hosts by including multivariable which are MIPS, RAM, bandwidth, and storage.

One of the reasons that lead to the improvement of the performance in PCA and clustering is the use of a dynamic algorithm. As mentioned in the paper, dynamic load balancing methods can overcome intrinsic changes in cloud environments. The second reason is the PCA and clustering model uses batch-processing to support long-term scheduling. The third reason is based on the main feature of our proposed model which is the use of a multivariate statistic to consider the cloud heterogeneous environment.

VI. CONCLUSIONS

This work proposes a new VM allocation policy that can dynamically allocate a new VM based on its required computing resources. This policy is added with multivariate analysis of the computing resources to extract the status of the hosts. The multivariate techniques which are PCA and clustering technique have been used to allocate the new VM with the most appropriate physical host. The results show that the performance of the load balancing for cloud computing environment has been improved in terms of Makespan and resource utilization. Incorporating this heterogeneous cloud resource into a VM allocation policy is crucial in revealing the signature of the computing resources to enable accurate and early allocation. The future suggestion for this work is to compare this policy with more allocation policies using other parameters like energy consumption and cost.

REFERENCES

[1] P. Mell, and T. Grance, "The NIST definition of cloud computing," *Technical Report No. SP 800-145*, The NIST Definition of Cloud Computing, National Institute of Standards & Technology, Gaithersburg, 2011.

[2] A. Shawish, and M. Salama, "Cloud computing: paradigms and technologies," in *Inter-cooperative collective intelligence: Techniques and applications*: Springer, pp. 39-67, 2014

[3] A. M. Kadhum, and M. K. Hasan, "Assessing the Determinants of Cloud Computing Services for Utilizing Health Information Systems: A Case Study," *International Journal o Advanced Science,*

Engineering and Information Technology, vol. 7, no. 2, pp. 503-510, 2017.
[4] K. Parikh, N. Hawanna, H. PK, and N. C. S. Iyengar, "Virtual machine allocation policy in cloud computing using cloudsim in java," *International Journal of Grid and Distributed Computing,* vol. 8, no. 1, pp. 145-158, 2015.
[5] P. Samimi, Y. Teimouri, and M. Mukhtar, "A combinatorial double auction resource allocation model in cloud computing," *Information Sciences,* vol. 357, pp. 201-216, 2016.
[6] N. Fonseca, and R. Boutaba, *Cloud Services, Networking, and Management*. John Wiley & Sons, 2015.
[7] L. S. Xue, N. A. A. Majid, and E. A. Sundararajan, "Quality of Service Evaluation of IaaS Modeler Allocation Strategies on Cloud Computing Simulator," in *Proceedings of the International Conference on High Performance Compilation, Computing and Communications*, pp. 6-10: ACM, 2017
[8] S.-C. Wang, K.-Q. Yan, W.-P. Liao, and S.-S. Wang, "Towards a load balancing in a three-level cloud computing network," in *3rd IEEE Conference on Computer Science and Information Technology [ICCSIT],* vol. 1, pp. 108-113, 2010
[9] P. Devi, and T. Gaba, "Implementation of cloud computing by using short job scheduling," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 3, no. 7, pp. 178-183, 2013.
[10] K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal, and S. Dam, "A genetic algorithm (ga) based load balancing strategy for cloud computing," *Procedia Technology,* vol. 10, pp. 340-347, 2013.
[11] R. Kaur, and P. Luthra, "Load Balancing in Cloud System using Max-min and Min-min Algorithm," *International Journal of Computer Applications (0975–8887),* 2014.
[12] D. B. LD, and P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Applied Soft Computing,* vol. 13, no. 5, pp. 2292-2303, 2013.
[13] D. Nusrat Pasha, A. Agarwal, and R. Rastogi, "Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment," *International Journal of Advanced Research in Computer Science and Software Engineeringl*, vol. 4, no. 5, pp. 34-79, 2014.
[14] S. Sethi, A. Sahu, and S. K. Jena, "Efficient load balancing in cloud computing using fuzzy logic," *IOSR Journal of Engineering,* vol. 2, no. 7, pp. 65-71, 2012.
[15] J. Zhao, K. Yang, X. Wei, Y. Ding, L. Hu, and G. Xu, "A Heuristic Clustering-based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no 2, pp. 305-316, 2016.
[16] Z. Á. Mann, "Allocation of virtual machines in cloud data centers—a survey of problem models and optimization algorithms," *ACM Computing Surveys (CSUR),* vol. 48, no. 1, pp. 1-34, 2015.
[17] A. Beloglazov and R. Buyya, "Energy efficient allocation of virtual machines in cloud data centers," in *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, 2010, pp. 577-578: IEEE.
[18] K. Garala, N. Goswami, and P. D. Maheta, "A performance analysis of load Balancing algorithms in Cloud environment," in *Computer Communication and Informatics (ICCCI), 2015 International Conference on*, 2015, pp. 1-6: IEEE.
[19] A. Jula, E. Sundararajan, and Z. Othman, "Cloud computing service composition: A systematic literature review," *Expert Systems with Applications,* vol. 41, no. 8, pp. 3809-3824, 2014.
[20] M. Nitika, M. Shaveta, and M. G. Raj, "Comparative analysis of load balancing algorithms in cloud computing," *International Journal of Advanced Research in Computer Engineering & Technology,* vol. 1, no. 3, pp. 120-124, 2012.
[21] B. Mondal, K. Dasgupta, and P. Dutta, "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach," *Procedia Technology,* vol. 4, pp. 783-789, 2012.
[22] G. Lee, "Resource allocation and scheduling in heterogeneous cloud environments," *Technical Report No. UCB/EECS-2012-78,* 2012.
[23] H. Abdi, and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics,* vol. 2, no. 4, pp. 433-459, 2010.
[24] F. Boobord, Z. Othman, and A. Abubakar, "PCAWK: A Hybridized Clustering Algorithm Based on PCA and WK-means for Large Size of Dataset," *Int. J. Advance Soft Compu. Appl,* vol. 7, no. 3, 2015.
[25] B. Gowri, R. Rajmohan, D. Dinagaran, M. Pajany, and A. Divya, "PCA based Lattice Clustering Methodology for Heterogeneous Dataset Modeling and Analysis," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 6, no. 3, pp. 78-83, 2016.
[26] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience,* vol. 41, no. 1, pp. 23-50, 2011.
[27] N. A. A. Majid, M. P. Taylor, J. J. J. Chen, W. Yu, and B. R. Young, "Diagnosing faults in aluminium processing by using multivariate statistical approaches," *Journal of Materials Science,* journal article vol. 47, no. 3, pp. 1268-1279, 2012.
[28] O. R. Zaïane, "Introduction to data mining," 1999, Available: http://webdocs.cs.ualberta.ca/~zaiane/courses/cmput690/notes/Chapter1/index.html