UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

A Cloud-based Conceptual Framework for Multi-Objective Virtual Machine Scheduling using Whale Optimization Algorithm

Nadim Rana, Muhammad Shafie Abd Latiff & Shafi'i Muhammad Abdulhamid
Faculty of Computing
Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Johor, Malaysia
&
Federal University of Technology Minna, Nigeria
Email: nadimrana17@gmail.com

Submitted: 12/01/2018. Revised edition: 12/06/2018. Accepted: 26/06/2018. Published online: 21 November 2018

***Abstract*—Virtual machine scheduling in the cloud is considered one of the major issue to solve optimal resource allocation problem on the heterogeneous datacenters. Concerning that, the key concern is to map the virtual machines (VMs) with physical machines (PMs) in a way that maximum resource utilization can be achieved with minimum cost. Since scheduling is an NP-hard problem, a metaheuristic approach is proven to achieve a better optimal solution to solve this problem. In a rapid changing heterogeneous environment, where millions of resources can be allocated and deallocate in a fraction of the time, modern metaheuristic algorithms perform well due to its immense power to solve the multidimensional problem with fast convergence speed. This paper presents a conceptual framework for solving multi-objective VM scheduling problem using a novel metaheuristic Whale optimization algorithm (WOA). Furthermore, we present the problem formulation for the framework to achieve multi-objective functions.**

***Keywords*—Cloud computing, VM scheduling, Metaheuristic, Whale optimization algorithm**

I. INTRODUCTION

Cloud computing is an on-demand computing model that provides distributed system services through the internet. The computing recourses are rationed to the clients over a virtualized network facility by cloud service providers [1, 2]. Cloud computing structure provides a stacks of provisioning systems, in which the traditional ones being: Infrastructure as a Service (IaaS) which deals with the services involving virtual resources, such as servers, storage, compute nodes (processors) and network bandwidth. Amazon is one of the prominent cloud providers that offers Infrastructure as a Service (IaaS) for vendors and users. For instance, the service module Elastic Compute Cloud (EC2) and Simple Storage Service (S3) are offered by Amazon for virtual infrastructure management. The second service model is called Platform as a Service (PaaS) that provides multiple flavors of operating systems and a lot of system tools and utilities as a service to the customers. It also allows computer programmers and engineers to develop and deploy their applications in the cloud. Microsoft Azure is an example of PaaS. The third model is the Software as a Service (SaaS) which permits suppliers to concede client’s access to authorized software. SaaS manages to utilize any application or administration through the cloud. Google calendar is one of the applications that gives coordinated effort on various applications, similar to event management, project management, etc., just by using a thin client. Salesforce is additionally a typical and well-known SaaS-based application of client relationship management (CRM) [3]. Each PM in cloud datacenter can holds several provisioned VMs which further consists of several virtual resources attached to it, such as CPUs, memory, storage, and network as shown in Fig. 1.

The organization of the paper is aligned as follows. A general description of virtualization is mentioned in Section 2. The VM scheduling and its problem formulation are discussed in Section 3, while the previous metaheuristic literature for solving VM scheduling problem is given in Section 4. In section 5, we propose WOA algorithm based conceptual framework. Finally, the discussion and conclusion are outlined in Section 6.

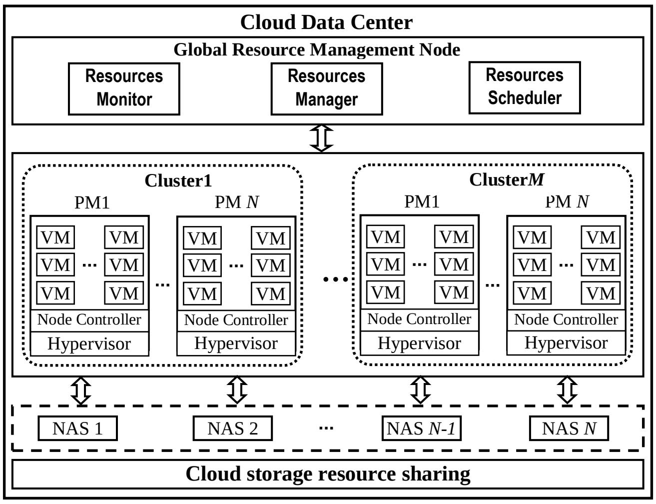

Fig. 1. Virtual Resource Model of Data Center [4]

II. CLOUD VIRTUALIZATION

Virtualization is a technique to create the VM with other facilities and resources like processor, storage, network, OS, file, memory which are consumed by the end user as a service through VM. VM is an emulation of a physical computer that runs OS and applications based on architecture and functionalities [4]. It is a software that runs the physical resources on a host with virtual devices to provide functionalities like portability, security, efficiency and reliability. The Virtual Machine Monitor (VMM) supports the execution environment and manages the VM resources such as policy-based automation, virtual hard disk, lifecycle management, live migration and real-time resource allocation [5].

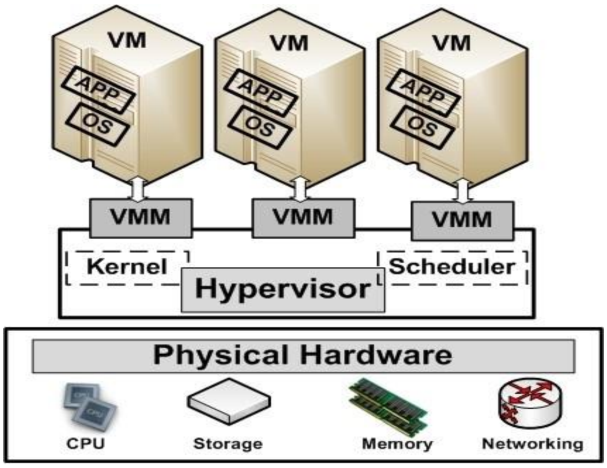

Fig. 2. Virtualized Environment Representations

There are two types of hypervisors in VM management namely, type I and type II. Type I hypervisor is used to establish the VM communication and hardware whereas type II hypervisor is used to run the VM on host operating system. The cloud virtualized setting comprises of the essential hardware, VMM, VMs, operating systems and softwares mounted on these VMs as shown in Fig. 2.

The VMM functions as a means of supervision application. It supervises the virtualized datacenter, facilitate to organize and monitor the virtualization host, networking equipments, and datacenter storage resources to produce and deploy VMs and services to the private clouds that have been produced [6]. Subsequently, these VMs can be quickly scheduled and freed with poor supervision work or insufficient facility provider interaction, resulting a quick ascent in the utilization of virtual resources in the datacenter. In this manner, it becomes a major issue to adopt a balanced resource management approach between the VMs and its datacenter resources to deliver better performance and services. Additionally, the consumer of the cloud and the larger part of users being served by application deployment in cloud provider's environment can achieve better SLA [7].

III. VIRTUAL MACHINE SCHEDULING PROBLEM

Virtual machine scheduling is a process of figuring out which movement ought to be performed as per the user demand. It is a request for the resources for an appropriate assignment of VMs and its resources to a PMs on a datacenter. The end goal of the VM scheduling is always be the maximum utilization of VMs resources with minimum incurred cost. Hence, efficient VM scheduling is always remains the key concern of cloud providers, and considered to be a primary feature of cloud computing [8, 9].

VM scheduling problem in cloud computing datacenter can be expressed as the distribution of a numbers of VMs to a numbers of PMs in a way that the amount of PMs can be minimized. In the study in [10], the authors propose a VM assignment in the cloud datacenter. The technique stressed up on the efficient VM provisioning to physical hosts to reduce the entire resource utilization and to minimize the number of PM required for a specific usage. The outcome of the simulation shows the lowest provisioning of the VMs to the host machine. The proposed algorithm outperforms in terms of its efficiency and scalability. However, the algorithm did not consider load balancing issues and prone to resource wastage at peak periods. In another similar work in [11], the researchers exploits the capability of ant colony optimization algorithm (ACO) for VM allocation problem. The authors perceived the combinatorial problem of bin packing as an NP-hard problem and modified the ACO by adding local search algorithm to optimize the allocation result. A comparison is made with the traditional best fit decreasing (BFD) algorithm with the proposed algorithm regarding resource utilization and power consumption.

A. Problem Formulation of VM Scheduling

1) VM Model

Fig. 3Fig 3 demonstrates the scheduling mechanism of VMs and PMs. The set of PMs is denoted by p= $\{P_1, P_2, \ldots,$

P_N }. N is referred as number of PMs on a datacenter, whereas the set of VMs allocated on PM is denoted by i. Suppose we need to deploy VM v at present, so to represent as a mapping solution set S = {$S_1, S_2, \ldots, S_N$} is used after v is scheduled to each PM. Here S_i referred as a set of mapping solution corresponding to VM v mapped to the PM P_i.

2) The Expression of Load

The load of a PM can be achieved by increasing the loads of the VMs running on it. In this regard, T is considered the best time span examined by historical data. Here, T is considered as the current time zone from the current time zone of monitoring historical data. As per the dynamic nature of PM load, we can divide time t into n time periods. Therefore we define $T = [(t_1 - t_0), [(t_2 - t_1) \ldots [(t_n - t_{n-1})]$.

As per the expression, $(t_k - t_{k-1})$ represents the time period k. Let's assume that the load of the VMs is somewhat stable in every period, then it may be define for the load of VM no. i for the period k is $v(i, k)$. Hence, it may be conclude that, for the cycle T and for the VM v_i, the average load on the PM p_i can be expressed as

$$\overline{v_i(i,\tau)} = \frac{1}{\tau}\sum_{k=1}^{n} v(i,k) \times (t_k - t_{k-1}) \qquad (1)$$

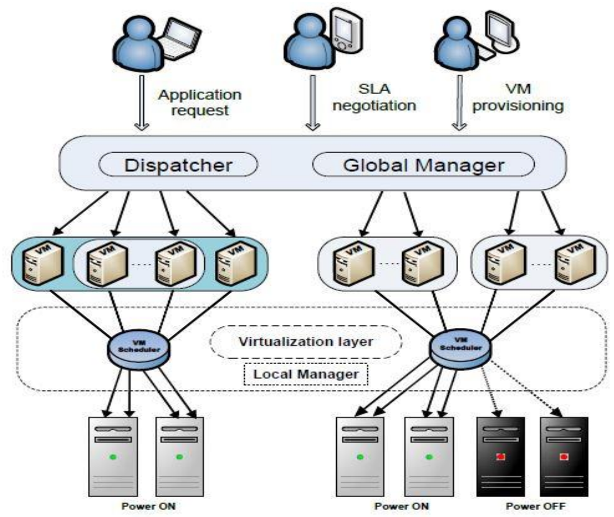

Fig. 3. Virtual Machine Scheduling Overview

As per the system configuration, The load of a PM can be calculated by increasing the loads of the VMs running on it. Thus the load on the PM p_i can be summarized as

$$\rho(i,T) = \sum_{j=1}^{m_i} \overline{v_i(j,\tau)} \qquad (2)$$

The deployment of the current VM is v. Since the VM has already configured with its required resources information, thus we can compute the current load of the VM as v' according to the appropriate information. Therefore, when VM v is allocated to a PM, the load of every PM is

$$\rho(i,\tau)' = \begin{cases} \rho(i,\tau) + v' & \text{After deploy } v \\ \rho(i,\tau) & \text{Others} \end{cases} \qquad (3)$$

Generally, when the VM v is allocated to the PM p_i, an obvious rise in the system load can be observed. So, we need to do an arrangement to load balance of PMs by distributing the workload with under loaded PMs. Thus, the load variation of the PMs can be expressed by the following expressions 4 and 5. Where mapping solution S_i in time t period after VM v is allocated to PM p_i is

$$\sigma_i(\tau) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\overline{\rho(\tau)'} - \rho(i,\tau)'\right)^2} \qquad (4)$$

Where

$$\overline{\rho(\tau)'} = \frac{1}{N}\sum_{i=1}^{N} \rho(i,\tau)' \qquad (5)$$

3) Mathematical model

According to the above expressions, we define the following model:

Lemma I: In the system representation of solution S_i, the load capacity of each PM is $P(i, t)'$, and the total load difference in time period t is expressed as

$$\sigma_i(S_i,\tau) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\overline{\rho(\tau)'} - \rho(i,\tau)'\right)^2} \qquad (6)$$

Where

$$\overline{\rho(\tau)'} = \frac{1}{N}\sum_{i=1}^{N} \rho(i,\tau)' \qquad (7)$$

Lemma II: The stable mapping solution of the structure representation where solution S_i is S'_i, and the sequence of representation solution S' must match to the sequence of stable representation solution S' = {$S'_1, S'_2, \ldots, S'_N$} is the best representation solution to form $\sigma(S_i, t)$ to fulfill the predefined load limitations.

Lemma III: The ratio of VM numbers M' that required to transfer to other PMs in order to attain load balancing in a definite representation solution to the overall VM(v) count M as cost divisor. Then for every representation solution S_i, the cost divisor $\delta(S_i)$ to reach load balancing S'_I is expressed as

$$\delta(S_i) = \frac{M'}{M} \tag{8}$$

IV. METAHEURISTIC APPROACH FOR VM SCHEDULING PROBLEM

Considering the system variation and historical data as a significant aspect of load balancing within the dynamic cloud environment, the authors in [12] develop an innovative technique by improving genetic algorithm for solving scheduling problem. The technique is designed to attain better load balancing of the PMs and to decrease dynamic migration of the VMs. The test results prove that the proposed algorithm performs better when compared with traditional algorithms to achieve better load balancing and low migration cost. Similarly, the VM scheduling can be defined as the allocation of a numbers of VMs to a numbers of PMs. In this regard, a particle swarm optimization (PSO) based improved VM scheduling policy is presented for VM allocation in cloud data centers [10]. The proposed policy intelligently allocates the VMs to physical hosts to cut short the total resource wastage and the numbers of servers used. Simulation results show that the proposed policy not only minimizes the allocation of VMs on the host machine but also achieves better performance and scalability.

In a virtualized cloud environment, the incoming request tends to be dynamic. The system that is responsible for creating the VM does not consider what types of requested tasks are going to execute on them. Hence, scheduling that considers the only limited amount of tasks or that needs the detailed information of the tasks is not applicable for this type of system. In another kind of research in [13], the authors present a load balancing aware hybrid metaheuristic algorithm combining the modern ACO with PSO to address VM scheduling issue. The proposed algorithm ant colony optimization with particle swarm (ACOPS) uses previously stored information on the server to predict the incoming workload in order to adjust to the ever-changing environments, and does not require additional task information. Moreover, ACOPS denies the request that does not fulfill the scheduling requirements to decrease the computing time. The simulation outcomes show that the presented algorithm outperforms when compare to the other benchmark techniques and also achieved reasonable load balancing. In a similar study, an improved method is anticipated based on ACO algorithm for VM allocation problem [11]. The authors perceived the combinatorial problem of bin packing as an NP-hard problem and modified the ACO by adding local search algorithm to optimize the allocation result. A comparison is done with the best fit decreasing (BFD) algorithm with the proposed algorithm concerning resource utilization and power consumption. The experimental results show better results were obtained.

V. PROPOSED WOA ALGORITHM

Whale Optimization Algorithm is a recently developed metaheuristic algorithm which is based on the bubble-net hunting maneuver of humpback whales. It is a swarm-based intelligent algorithm designed to solve a complex engineering optimization problems in the continuous domain. The WOA has been widely used and tested algorithm to solve optimization problems in multidisciplinary areas, such as electrical and power systems, computer science, aeronautical and navigation systems, Wireless networks, construction and planning are amongst few of them [14]. Furthermore, several works can be seen in the literature for the improvement of the basic WOA in terms of modification, hybridization, and multi-objective function in various fields and sub-fields. The proposed WOA have been tested on 29 mathematical functions and 6 benchmark structural engineering problems to test its validity. As per the result, it has shown a tremendous capability in terms of efficiency and robustness for solving optimization problems [15]. Due to its simple structure, it allows the researchers to improve the algorithm just by fine tune a single parameter (time interval). In the herding process, the swarm of the whale searched through a multidimensional space for their prey, in that space the location of the each whale is represented as different decision variable, while the distance between the each whale with respect to the food is measured as objective cost. The time-constraint based position of the individual whale is measured by three different operational processes: (i) shrinking encircling prey, (ii) bubble-net attacking method (exploitation phase) and (iii) search for prey (exploration phase). Fig. 4-7 show the basic representation of the WO algorithm processes. The mathematical expressions of these operational processes are mentioned in the following subsections:

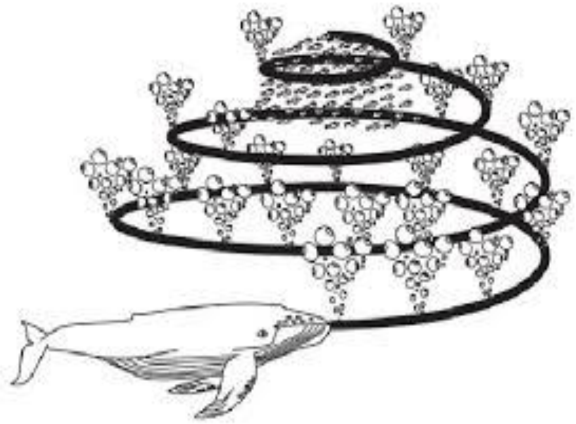

Fig. 4. Bubble-net Feeding of Humpback Whales

The unique hunting maneuver of the Humpback whales is used to identify the target prey location. Once the target prey is identified by the search agents, they start to encircle them moving around in 9 shapes as shown in Fig. 5.

Because the optimal position of the prey is not known in advance in the vast search space, the WOA presumably considers the target prey as the current best candidate solution or near to optimum solution. All the possible effort is carried out to find the best search agent in the exploration and exploitation phase by the WOA, whereas the other search agents update their positions near to the best search agent. The searching and hunting behavior of the humpback whale is mathematically modeled in [15].

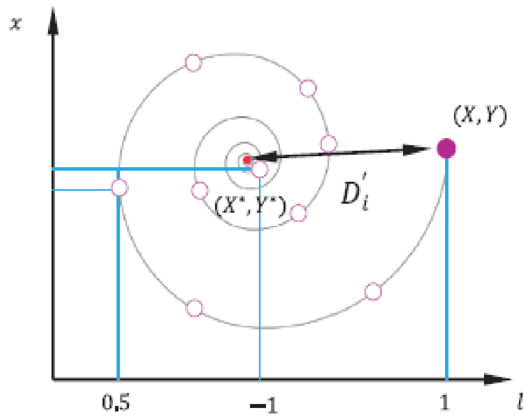

Fig. 5. Spiral Updating Position

$$\vec{D} = |C.\overrightarrow{X^*}(t) - \vec{X}(t)| \quad (1)$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - A.\vec{D} \quad (2)$$

$$A = 2.a.r - a \quad (3)$$

$$C = 2.r \quad (4)$$

Where $\overrightarrow{X^*}$ is the general best position, $\vec{X}$ represent whale position and *t* specifies the recent iteration, *a* represent linearly reduced within the range of 2 to 0 over the period of iterations, and *r* is a random number equally distributed in the range of [0, 1].

A. Bubble-net Attacking Method (Exploitation Phase)

The representation of the bubble-net hunting behavior which is called exploitation phase of the WOA as shown in Fig 6. A spiral mathematical formulation is applied to identify the equidistance between the whale position and the position of the prey. After each movement, an updates is performed between the whale position in a helix shape in order to adjust with current search space [16]:

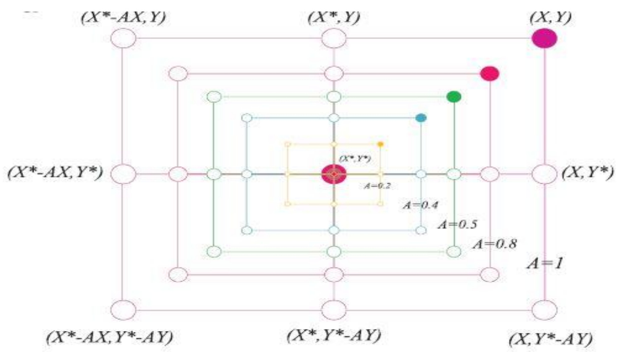

Fig. 6. Exploitation Mechanism (X * is the best solution obtained so far)

$$\vec{X}(t+1) = e^{bk}.\cos(2\pi k).\vec{D'} + \overrightarrow{X^*}(t) \quad (5)$$

$$\vec{D'} = |\overrightarrow{X^*}(t) - \vec{X}(t)| \quad (6)$$

In the given expression, b represents a constant for explaining the logarithmic spiral shape whereas k is an arbitrary number equally distributed in the range of [-1, 1].

B. Search for Prey (Exploration Phase)

To have global optimizers, if A>1 or A <-1, the search agent is updated as stated by a randomly chosen search agent in the place of the best search agent:

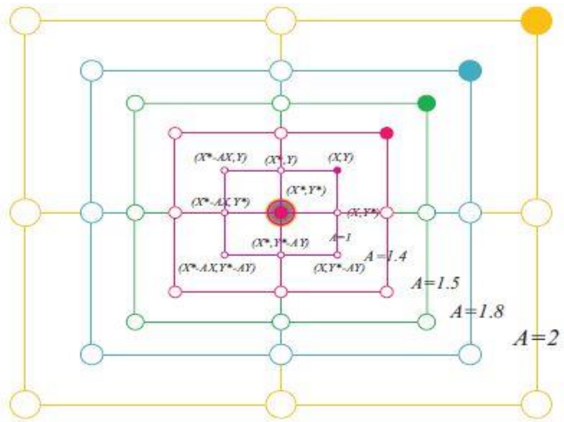

Fig. 7. Exploration Mechanism (X ∗ is a randomly chosen agent)

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - A.\vec{D}' \quad (7)$$

$$\vec{D}'' = |C.\overrightarrow{X_{rand}} - \vec{X}(t)| \quad (8)$$

Where, $C.\overrightarrow{X_{rand}}$ is nominated arbitrarily from whales in the current iteration. For further details, the reader may refer to [15].

Pseudo-code 1: Basic Whale optimization algorithm

```
Initialize the whales population Xi(i = 1, 2, …, n)
Calculate the fitness of each search agent
X* = the best search agent
while(t < maximum number of iterations)
    for each search agent
        Update a, A, C, l, and p
        if 1(p<0.5)
            if 2(|A|< 1)
                Update the position of the current search agent by the Eq.(1)
            else if 2(|A|>1)
                Select a random search agent ( )
                Update the position of the current search agent by the Eq. (8)
            end if 2
        elseif 1(p>0.5)
                Update the position of the current search by the Eq. (5)
        end if 1
    end for
    Check if any search agent goes beyond the search space and amend it
    Calculate the fitness of each search agent
    Update X* if there is a better solution
    t=t+1
end while
return X*
```

VI. CONCLUSION AND FUTURE WORK

The paper provides a conceptual framework for solving VM scheduling problem in a heterogeneous environment. Scheduling problem in the cloud is considered as NP-hard, so a metaheuristic WOA based framework is presented to address this problem. We anticipate that, by exploiting the unprecedented features of the proposed algorithm, such as robustness, fast convergence rate, and local optima avoidance capability, the framework will optimally schedule the VMs resources on the best suitable PMs. We also anticipate that the WOA algorithm will perform well in solving multi-objective problems like makespan, resource utilization and load balancing.

The future direction of research could be the extension of the proposed algorithm with its modification and hybridization with other methods. Also, the proposed algorithm can be utilized to address various types of objective functions such as energy, throughput, response time, execution time, bandwidth, cost, SLA voidance, fault tolerance etc. for the purpose of cloud scheduling.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the Pervasive Computing Research Group for conducting this research in Faculty of Computing at Universiti Teknologi Malaysia.

REFERENCES

[1] Buyya, R., Broberg, J., & Goscinski, A. M. (Eds.). (2010). *Cloud Computing: Principles and Paradigms.* John Wiley & Sons, Vol. 87.

[2] Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2012). Cloud Computing and Grid Computing 360-Degree Compared. IEEE Grid Computing Environments (GCE08) 2008, co-located with IEEE/ACM Supercomputing 2008. *2012 ACM/IEEE 13th International Conference on Grid Computing*.

[3] Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008, November). Cloud Computing and Grid Computing 360-degree Compared. *Grid Computing Environments Workshop, 2008. GCE'08*. Ieee, 1-10.

[4] Ghiasi, H., & Arani, M. G. (2015). Smart Virtual Machine Placement Using Learning Automata to Reduce Power Consumption in Cloud Data Centers. *SmartCR*, 5(6), 553-562.

[5] Xu, F., Liu, F., Liu, L., Jin, H., Li, B., & Li, B. (2014). iaware: Making Live Migration of Virtual Machines Interference-aware in the Cloud. *IEEE Transactions on Computers*, 63(12), 3012-3025.

[6] Eden, M. (2011). A survey of Performance Modeling and Analysis Issues in Resource Management across x86-based Hypervisors in Enterprise Cloud Computing Deployments.

[7] PushpaLatha, K., Shaji, R. S., & Jayan, J. P. (2014). A Cost Effective Load Balancing Scheme for Better Resource Utilization in Cloud Computing. *Journal of Emerging Technologies in Web Intelligence*, 6(3), 280-290.

[8] Achar, R., Thilagam, P. S., Shwetha, D., & Pooja, H. (2012, November). Optimal Scheduling of Computational Task in Cloud Using Virtual Machine Tree. *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*. IEEE, 143-146.

[9] Rana. N., Latiff M. S. A., Abdulhamid S. M., and Chiroma H. (2018). Virtual Machine Scheduling Techniques in Cloud Computing: Systematic Literature Review and Future Trends, *in press. Cluster Computing.*

[10] Kumar, D., & Raza, Z. (2015, February). A PSO based VM resource Scheduling Model for Cloud Computing. *Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on*. IEEE, 213-219.

[11] Gondhi, N. K., & Sharma, A. (2015, May). Local Search Based Ant Colony Optimization for Scheduling in Cloud Computing. *Advances in Computing and Communication Engineering (ICACCE), 2015 Second International Conference on*. IEEE, 432-436.

[12] Hu, J., Gu, J., Sun, G., & Zhao, T. (2010, December). A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment. *Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on*. IEEE, 89-96.

[13] Cho, K. M., Tsai, P. W., Tsai, C. W., & Yang, C. S. (2015). A Hybrid Meta-Heuristic Algorithm for VM Scheduling with Load Balancing in Cloud Computing. *Neural Computing and Applications*. 26(6), 1297-1309.

[14] Rana. N., Latiff M. S. A., Abdulhamid S. M., and Chiroma H. (2018). Nature-inspired Metaheuristic: Developments on the Modification of Whale Optimization Algorithm and its Applications, *Swarm and Evolutionary Computation* unpublished.

[15] Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51-67.

[16] Kaveh, A. (2017). Sizing Optimization of Skeletal Structures Using the Enhanced Whale Optimization Algorithm. *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*. Springer, Cham, 47-69.