

# Energy Efficient Virtual Machine Consolidation Using Water Wave Optimization

Rambabau Medara  
Computer Science and Engineering  
Indian Institute of Technology BHU  
Varanasi, India  
medararambabu.rs.cse18@iitbhu.ac.in

Ravi Shankar Singh  
Computer Science and Engineering  
Indian Institute of Technology BHU  
Varanasi, India  
ravi.cse@iitbhu.ac.in

Selva Kumar U  
Computer Science and Engineering  
Indian Institute of Technology BHU  
Varanasi, India  
selva.kumaru.cse16@itbhu.ac.in

Suraj Barfa  
Computer Science and Engineering  
Indian Institute of Technology BHU  
Varanasi, India  
suraj.barfa.cse16@itbhu.ac.in

**Abstract**—With the unprecedented growth of Cloud Computing, data centers around the world have increased exponentially. The energy utilization in these data centers is becoming a concern, so the need for energy-efficient algorithms in the cloud has been on the top agenda for quite a while. Cloud providers use different energy management strategies to minimize energy utilization and to maximize ROI (Return On Investment) such as energy-efficient virtual machine (VM) placement. We have taken the problem for VM consolidation and worked on it for an energy-efficient algorithm. VM Consolidation uses live migrations of VMs during the execution of cloudlets so that underloaded physical servers can be switched off by transferring those VMs to other physical machines. VM consolidation in the cloud environment is a proven NP-hard problem. We have used Water Wave Optimization (WWO) which is a meta-heuristic algorithm. Original WWO proposed for continuous space and so we have modified the parameters of the algorithm to apply for our problem of VM consolidation. Our approach produces a near-optimal solution using an objective function that minimizes energy consumption and increases the number of switched off servers.

**Index Terms**—cloud computing, Scheduling, energy-efficient, virtual machine consolidation and VM migration

## I. INTRODUCTION

Cloud computing depends on shared computing resources instead of having on-premise servers to deal with user applications. In other words, cloud computing is taking services and moving them outside an organization's firewall. The computing resources are delivered through the Internet and users are allowed to pay as-needed business model [1]. The important features of cloud computing are [2]:

*Self Service*: The customers can provide themselves with the required computing abilities such as processor time, network bandwidth without human intervention of cloud providers.

*Broad Network Access*: The customers of the clients of cloud can access the services by the client using cloud by the usage of the internet from any of the broad range of devices like laptops, smartphones, and tablets.

*Resource Pooling*: The resources such as processor, memory, network bandwidth given by the cloud provider is shared using the multi-tenant model where multiple users share resources from the same physical data centers. But independence is provided by a level of abstraction called virtualization.

*Rapid Elasticity*: One of the great advantages of the cloud is that the client companies need not worry about their increase in customers and hence, increase in the traffic of their services. Cloud provides seamless scaling mechanisms, thereby mitigating unforeseen circumstances.

*Measured Service*: The statistics of data and usage of services in the cloud can be easily collected from the frameworks of cloud, thereby helping companies to use that valuable data to increase their revenue and growth.

### A. Virtualization

The process of creating virtual one in place of actual things such as operating system (OS), network resource, a memory system or a server is called virtualization. It has been established for decades for improving resource utilization and sharing [3]. It was extensively used mainframe systems to improve manageability, reliability and resource utilization [4]. The capability of virtualization is broadly used in workload consolidation, workload isolation and workload migration [5].

In the case of Cloud Computing, Virtualization technique does a great deal of cost-saving, energy-saving, hardware performance enhancement for the cloud providers. It allows a single Physical Machine (PM) of a data center to be shared among multiple applications of multiple users at the same time. Virtualization is achieved by assigning a logical name to physical resources and providing a pointer to that resource on a need basis. Virtualization not only provides a way to execute multiple shared applications but also helps in sharing hardware resources of processor, memory and network Bandwidth. Virtualization in cloud computing is done by a software called Hypervisor or Virtual Machine Monitor (VMM) which creates

an abstraction layer between the software and the hardware in use. In this system, a physical machine (PM) can be virtualized into multiple instances at a time, but every instance logically isolated from each other for security reasons.

### B. VM Consolidation

The various virtual machines (VM) running on different physical machines of a datacenter need to get processor time, memory and bandwidth for the completion of tasks assigned to it. The VMM or hypervisor is responsible for the allocation of VMs on the PMs and this is done by the various Virtual Machine Allocation algorithms which are available. We concern ourselves with the energy-efficient allocation algorithms and moreover the ones which do dynamic migrations of VMs during Runtime. The process in which dynamic migration of the VMs happens during runtime to reduce a load of overloaded Physical Machines (PM) and to put to sleep the underloaded PMs is called as VM Consolidation. Switching off the underloaded hosts will help us reduce the energy consumption of the data centers and this can be done by migrating the VMs from those machines to other machines [6]. VM Consolidation with energy-aware objective is an NP-Hard problem [7-8][10] which leads us to the use of meta-heuristic algorithms for the same. One such meta-heuristic algorithm is Water Wave Optimization (WWO) which is inspired by the characteristics of waves in seas and oceans. The original WWO algorithm was proposed for continuous space multidimensional optimization problem. Later, it was adapted to solve the famous discrete optimization problem of Travelling Salesman Problem (TSP) [9] by modifying its original parameters. This work motivated us to try the WWO by modifying the parameters and apply to our multidimensional discrete optimization problem of Energy Efficient VM Consolidation.

The rest of this article organized as follows. We introduce the related work in Section II. The Cloud Data Center (CDC) model discussed in Section III. The overview of the water wave theory introduced in Section IV. The proposed WWO-based VM Consolidation algorithm presented in Section V. Section VI presents the experimental setup and discussion on results. Finally Section VII concludes our work and discussed the future directions.

## II. RELATED WORK

There have been many works done to minimize energy consumption of data centers by maximizing the resource utilization. By applying the Ant Colony Optimization (ACO) algorithm, an approach [10] resolves the Multi-dimensional Bin Packing (MDBP) problem by effectively consolidating the VMs in the PMs through workload placement in a cloud environment. Despite, Energy-aware VM placement scheduling approach [11] measures the fitness value between VMs using an ACO algorithm to identify the past optimal placement in the corresponding PM rather than measuring the fitness between the VMs and PMs. It tends to provide the solution with a minimum number of PMs, which helps to reduce the overall energy consumption. ACO metaheuristic based scheduling

method [12] consolidates the VMs by incorporating the vector algebra, which minimizes the energy consumption and reduces the resource wastage in the cloud environment. A genetic algorithm also has been used by the existing researchers to develop the energy-efficient cloud datacenter through VM consolidation [13].

To reduce the energy consumption through minimizing the time, memory, and cost consumption, Task-based load balancing approach [14] employs the Particle Swarm Optimization (PSO) algorithm to migrate only several tasks to the identical VM resource rather than migrating the entire overloaded VM. A modified PSO algorithm [15] ensures the energy-efficient VM placement in the cloud data center by optimizing the operators and parameters of the PSO algorithm. In order to minimize the energy utilization in the workflow scheduling, the recent research works have focused on modeling the energy-aware scheduling algorithms through the resource hibernation, dynamic power management, or Dynamic Voltage and Frequency Scaling (DVFS) techniques. Multi-objective Discrete PSO (MODPSO) approach [16] employs the hybrid PSO algorithm and DVFS technique to reduce the energy utilization by the cloud infrastructure, which averts the compromise between the performance and energy consumption by handling multiple QoS requirements. Greedy based heuristics proposed in [28] which set upper and lower utilization threshold for Central Processing Unit (CPU) utilization. If exceed the upper threshold or utilization drops below the lower threshold then VM migration plan enforced.

Zheng Yu-Jun [19] proposed a new optimization technique called Water Wave Optimization (WWO) which is a metaheuristic inspired by shallow water wave models. WWO approach is competitive with few state-of-the-art evolutionary algorithms such as ACO, BA, BBO, IWO etc., and effective for real-time applications[22]. In this paper, we used WWO algorithms to effectively place virtual machines on a selected physical machine in an energy-efficient way while satisfying Qulaity of Service (QoS) requirements.

## III. SYSTEM MODEL

A heterogeneous Cloud Data Center (CDC) contains  $m$  physical machines (PMs). Each PM characterized by computing resources such as CPU, memory, storage capacity and network I/O. The performance of CPU defined in terms of Millions of Instructions Per Second (MIPS). The PMs are virtualized to serve many users at any given time. Users present their request for provisioning of  $v$  virtual machines of PM. The length of the user request specied in Millions of Instructions (MI). We used the greedy based approach called Best Fit (BF) for the initial allocation of VMs to PM. The BF algorithm is a well-known heuristic for bin-packing problem [17].

The VM resources utilization changes with time due to dynamic workloads. Hence, the initial provision of VMs to PM needs to be enhanced with an efficient VM consolidation approach. Our proposed WWO-ACS algorithm is applied periodically to optimize the VM placement depending on the

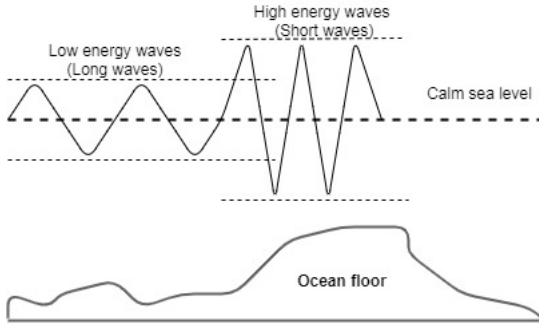


Fig. 1 Shallow and deep water wave models

workload. The BF algorithm used to optimize the resource utilization locally, whereas our proposed WWO-VMC approach used to optimize resource utilization globally. Based on the percentage of CPU utilization we categorize the PM into one of the three categories  $PM_{under}$ ,  $PM_{over}$  and  $PM_{normal}$ . We consider PM as underutilized  $PM_{under}$  if the CPU utilization under 40%, PM as over-utilized  $PM_{over}$  if CPU utilization over 90%, and all other PMs are normal utilized  $PM_{normal}$ .

#### IV. THEORY OF WATER WAVE

Isaac Newton was the first person to make an effort on the theory of water waves. Further, the linear theory of water waves was studied by French mathematicians Laplace, Lagrange, Poisson, and Cauchy and they made authentic theoretical advances [18]. The WWO approach proposed in [19] is an efficient technique for global optimization. The shallow water wave theory is the basis for WWO, which uses a numerical approach to analyze the evolution of wave amplitudes (heights), periods (wavelengths), and propagation directions under different conditions such as nonlinear wave interactions, wind force, and frictional dissipation [21-22]. The solution space in the WWO algorithm is comparable to the seabed region. The fitness of any point in the solution space is estimated inversely by the depth of the seabed. The fitness functions of wave measure high if the distance is less to the still water [19][22]. The various changes in the shape of water waves such as amplitude or height and wavelength depicted in Fig. 1.

The high energy waves having large amplitude i.e., good wave produce a high-quality solution and low energy waves having long wavelength produce poor solutions. The majority of evolutionary algorithms keep-up a population of solutions; similarly, WWO maintains a solution. Each solution of which is similar to a “wave” with two parameters height and wavelength. The height of wave  $h \in Z^+$  (integer domain) and wavelength  $\lambda \in R^+$  (real domain). These parameters are initialized as constants  $h = h_{max}$  and  $\lambda = 0.5$  [16]. The best solution exploration procedure in the WWO algorithm is modeled as the wave propagation, wave refraction, and wave breaking.

#### V. ENERGY EFFICIENT VM CONSOLIDATION WITH WWO

As discussed in Section III, each PM provides one or more VMs and both PMs and VMs categorized by their resource utilization. In the context of VM migration a PM is potential source PM  $pm_{source}$  where VMs already residing or target PM  $pm_{dest}$  for VM migration. All  $PM_{under}$  and  $PM_{over}$  are members of  $pm_{source}$ , where as all PMs except  $PM_{over}$  are members of  $pm_{source}$ . The proposed WWO-VMC algorithm creates tuples with three elements each  $t = (pm_{source}, vm, pm_{dest})$  where  $pm_{source}$  is the source host,  $vm$  is the selected VM for migration and  $pm_{dest}$  is destination host machine. Our proposed algorithm aims to minimize the number of active PMs needed to host all VMs without compromising their performance. This can be done by enforcing the VM consolidation algorithm.

In this work, we redesign the original WWO algorithm for VM Consolidation to minimize energy consumption. As every solution in WWO is a migration plan or set of migrations, hence, we considered every solution as a wave  $x$ . A wave has three parameters tuple  $t$ , height  $h$  and wavelength  $\lambda$  (initially height is constant  $h_{max}$  and wavelength is 0.5). The fitness of  $x$  depends on the number of migrations (M) and the number of sleeping hosts or PM (Ps): As the number of migrations increase, fitness decreases and as the number of sleeping hosts increases, fitness also increases [23].

We initialize the population as a set of migration plans as in the original WWO algorithm and its use for the three operators of wave propagation, refraction and breaking. Which evolve the population up to the termination state is satisfied continually. However VM consolidation is a combinatorial problem, therefore WWO cannot be directly applied for energy-efficient VM Consolidation, so we have to redefine its operator’s propagation, refraction and breaking as discussed in the following sections.

##### A. Propagation for Energy Efficient VMC

In the propagation operator, we can see wavelength  $\lambda$  of solution as the probability for mutation, and we will decide whether to mutate or not using: for a good solution and small probability of mutation the value of  $\lambda$  will be small; on the other hands, a bad solution will have a large  $\lambda$  value, and hence will have a considerable probability of being mutated.

Each wave  $x$  generates another wave  $x'$  if its wavelength is greater than a randomly generated value  $r$  between 0 and 1, by this method –

If  $r < \lambda$  then for each tuple  $t$  in wave  $x$ , we will randomly either add a new tuple or replace the current tuple (with a tuple from the set of all tuples excluding the tuples from current wave  $x$ ) or we will remove the current tuple from current wave  $x$ ; If the current wave  $x$  fitness is less than the fitness of the newly generated wave  $x'$  then replace  $x$  by  $x'$  (with  $h = h_{max}$ ) else set  $h = (h - 1)$  for current wave  $x$ . After each generation,  $x$  updates its wavelength as

$$\lambda = \lambda \cdot \alpha^{-(f(x)-f_{min}+\epsilon)/(f_{max}-f_{min}+\epsilon)} \quad (1)$$

where,  $f(x)$  is fitness function,  $f(x) = P_s^\gamma + 1/(\epsilon + M)$ ,  $P_s$  is the number of sleeping hosts(PM),  $M$  is the number of migration,  $\gamma$  is a parameter defining the relative importance of  $P_s$ ,  $\alpha$  is wavelength reduction coefficient,  $f_{min}$  and  $f_{max}$  are minimum and maximum fitness of the current solution (Migration plan). To avoid division by zero, we choose a small value for  $\epsilon$ .

### B. Refraction for Energy Efficient VMC

The purpose of Refraction in WWO is if we know that a solution is bad, then we try to improve the solution by giving it some good features of the known best solution. Technically If height  $h$  of some wave/solution  $x$  becomes zero, then  $x$  is replaced by new solution  $x'$  which will be centered between  $x$  and global best solution  $x^*$ .

In our case, our solution is the migration plan having some tuples showing each migration. So whenever the height of the solution vanishes, to make the current solution (migration plan) better, we add or replace some tuple (migrations) with tuples from the global best solution so far i.e.  $x^*$ .

We set the height of the new solution to  $h_{max}$ , and update the wavelength :

$$\lambda' = \lambda \cdot \frac{f(x)}{f(x')} \quad (2)$$

where  $f(x)$  and  $f(x')$  is the fitness of old solution and new solution respectively.

### C. Breaking for Energy Efficient VMC

If the propagated solution is better than the known global best solution so far, i.e.,  $f(x) > f(x^*)$ , then we check in the neighborhood of the solution  $x$  for a better solution than  $x$ , and if we find a solution better than  $x$  say  $x'$ , then the global best solution is updated as  $x^* = x'$  otherwise update  $x^* = x$ .

We will find the neighborhood solution and update the global best solution using the following way: For  $k$  in  $(1, k_{max})$  do: Choose some tuples from solution  $x$ , and for each of those chosen tuple either add or replace the tuple with a tuple from the set of all tuples excluding the tuples from current wave  $x$ , or remove the tuple from solution  $x$ . if  $f(x) > f(x^*)$  then update  $x^* = x$ . where  $k_{max}$  is a predefined number whose value is 12 [19].

### D. Algorithm framework

In this section, we presented our proposed algorithm and discussed time complexity. The main focus of our algorithm is to maximize the inactive machines, in other words, minimize the active machines to conserve energy. The pseudo-code for the proposed WWO-VMC algorithm given in Algorithm 1. It initializes the population randomly (line 1) and creates tuples with three elements each  $t = (pm_{source}, vm, pm_{dest})$  where  $pm_{source}$  is the source host,  $vm$  is the selected VM for migration and  $pm_{dest}$  is destination host machine. The number of tuples keeps changing according to the algorithm whenever add or remove the tuple step comes. The three operators used in this algorithm are propagation, refraction,

and breaking. Propagation (lines 5-14) enables to accept tuple or create new tuple based on the fitness of the population. For better wave propagation we consider parameters  $r_1$  and  $r_2$ . Based on the series of preliminary experiments we tuned  $r_1$  and  $r_2$  to 0.28 and 0.64 respectively. The breaking operator (line 17-19) enables an exhaustive search for selecting a source for migration. Finally the refraction (lines 21-24) avoids early convergence by improving population diversity.

---

#### Algorithm 1: WWO-VMC algorithm

---

```

1  Randomly initialize a population  $P$  of migration
   plans;
2  while  $iterations < max\_number\_of\_iterations$ 
   do
3    for each wave  $x$  in population  $P$  do
4      initialize  $x' = x$ 
5      for each tuple  $t$  in  $x$  do
6         $r = rand(0, 1)$ 
7        if  $rand() < x.\lambda$  then
8          if  $r < r_1$  then
9            add new tuple in  $x'$ 
10         else if  $r < r_2$  then
11           replace  $t$  by new tuple in  $x'$ 
12         else
13           remove  $t$  from  $x'$ 
14         end for
15       end for
16       if  $f(x') > f(x)$  then
17         if  $f(x') > f(x^*)$  then
18           perform breaking operation (section D)
19           update  $x$  with  $x'$ 
20         else
21            $x.h = x.h - 1$ 
22         if  $x.h == 0$  then
23           perform refraction operation (section B)
24           update wavelength based on equation (1)
25       end while
26     return  $x^*$ .
```

---

Our proposed WWO-VMC algorithm framework is clear and easy to understand. The algorithm performs well with fewer populations. We consider the recommended values of control parameters for the algorithm such as the maximum wave height  $h_{max}$ , the number of breaking directions  $k$ , wavelength reduction coefficient  $\alpha$  as in [19]. The WWO-VMC algorithm minimizes the number of active PMs to reduce the energy consumption of data centers, while preserving the QoS requirements.

Let  $n$  be the number of waves in the population,  $m$  be the number of physical machines (PM),  $v$  be the number of Virtual Machines (VM). Our algorithms have three types of operation in a single iteration over the population. Propagation: Time complexity of propagation is linear in the number of tuples in  $n$  waves of the population. The worst case of number of the tuples in a wave is  $O(m^2 * v)$  which implies the worst-case time complexity of this operation is  $O(n * m^2 * v)$ . This is

TABLE I: Parameters in the WWO-VMC approach.

$\gamma$	$\epsilon$	$r$	$\alpha$	$\lambda$	$\epsilon$	$k$
5	0.00001	(0, 1)	(1.001, 1.01)	0.5	0.00001	(1, 12)

the costliest operation in a single iteration. Breaking: Let  $k$  be the coefficient of breaking. The time complexity of breaking is  $O(k*m^2*v)$ . Refraction: It has the same worst-case complexity as that of propagation operation. Let  $I$  be the number of iterations in the algorithm. So, the overall time complexity of WWO-VM Consolidation algorithm is  $O(I * n * m^2 * V)$

## VI. EXPERIMENTAL SETUP AND RESULTS

The technology stack used for the simulation of results is JAVA language and the framework used was CloudSim Plus. Moreover, our target is IaaS clouds which provides unlimited resources to the cloud users on a payment basis. Conducting repeatable experiments on such infrastructure is expensive, so we used the simulation model. CloudSim Plus [24] is a toolkit with a full-featured and flexible simulation framework. It enables users to model cloud scheduling applications for simulation, and experimentation. Users are allowed to focus on specific system modeling issues to be explored, without regarding the functional level characteristics related to data center infrastructure and Services. CloudSim Plus framework has all the basic classes required for simulation of various processes of cloud. It allows one to extend the basic classes to implement the modified algorithms for the specific simulation. We have extended the basic VM allocation with migration to implement our WWO based VM allocation with migration. For the simulation of our algorithm, we have used the open data, provided by Numerical Aerodynamic Simulation (NAS) Systems Division at NASA Ames Research Center. The workload was logged for three months from October 1993 in a 128-node iPSC/860 hypercube [25].

We used the cleaned and converted log in Standard Workload format [26]. SwfWorkloadFileReader class in Cloudsim Plus was used to read and build cloudlets using this file. Each cloudlet was assigned a suitable VM, and correspondingly hosts were created according to requirements of VMs at initial simulation. For our hosts, we selected one HP ProLiant ML110 G3 server (with the configuration of (1 x [Pentium D930 3000 MHz, 2 cores], 4GB)). To evaluate the performance of our implemented algorithm, we considered two metrics energy consumption and the number of migrations. The WWO parameters and values which we were obtained in a series of preliminary experiments and that were used in the proposed WWO-VMC approach present in Table I.

### A. Simulation Results

The energy consumed is the whole energy utilized by the physical machines while running the VMs during the simulation. The objective function was designed in such a way that energy consumption is minimized by our algorithm. The energy consumed by a single physical machine depends on the utilization of resources of a machine like CPU, memory,

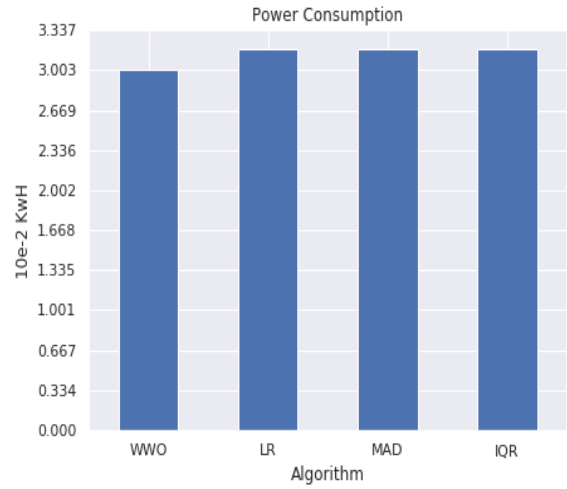


Fig. 2 15 Cloudlets

and bandwidth. It has been noted that power consumption by the utilization of CPU exceeds the other factors by a margin. So most of the approaches to calculate energy are based on modeling of energy based on the utilization of CPU. The SPECpower benchmark [27] is the real-world workload that we used to evaluate our algorithm on CloudSim Plus. The energy consumption of HP G3 server at different workload levels illustrated in Table II.

TABLE II: Amount of energy consumption of HP G3 server at different load levels.

Load Levels(%)	Energy Consumption(Watts)
0	105
10	112
20	118
30	125
40	131
50	137
60	147
70	153
80	157
90	164
100	169

We have compared our WWO-VMC algorithm with three well known heuristic methods for dynamic VM reallocation in [28]. These algorithms keep and use CPU utilization between upper and lower thresholds. When a PM underutilized then its VMs are consolidated for load balancing and when the PM overutilized (exceeds a threshold) then its VMs are reallocated for load-balancing. To estimate the PM utilization these heuristics adapt the utilization threshold dynamically based on LR (Local Regression), MAD (Median Absolute Deviation) and IQR (Interquartile Range).

We used different workloads to evaluate the performance of our proposed approach. The simulation results are depicted in Figs. 2-7 and it is clear that results show that our proposed WWO-VMC approach surpassed the other algorithms in energy saving. And a significant energy saving observed for large workloads.

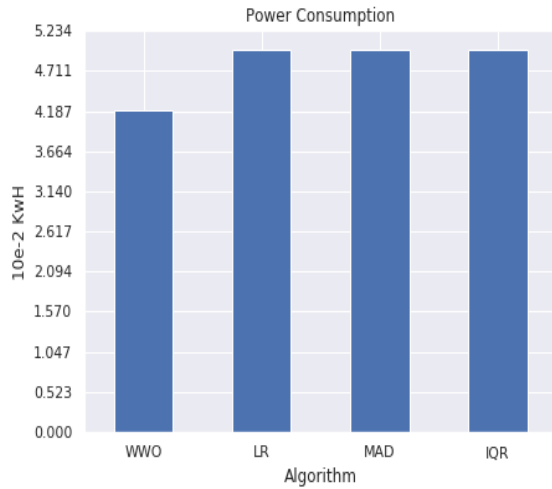


Fig. 3 20 Cloudlets

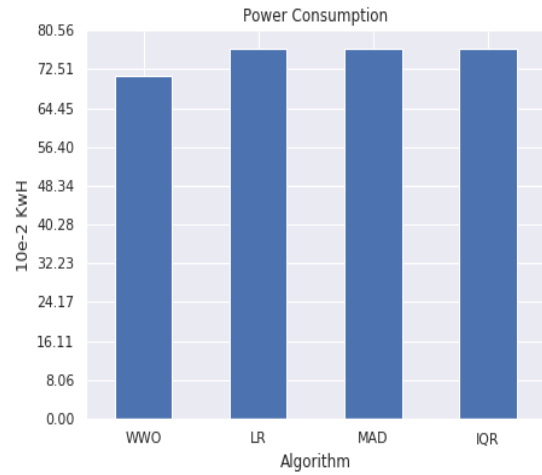


Fig. 6 90 Cloudlets

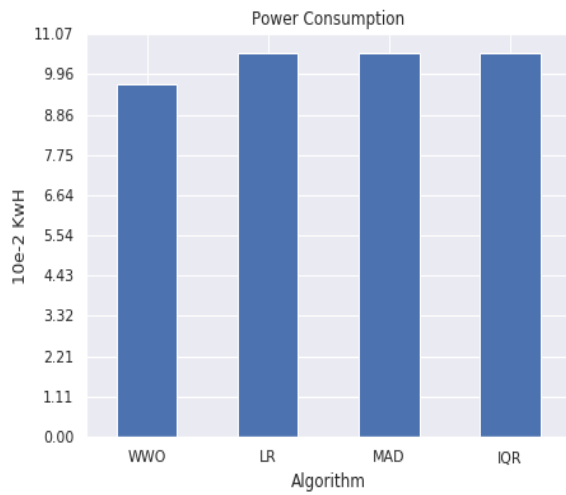


Fig. 4 30 Cloudlets

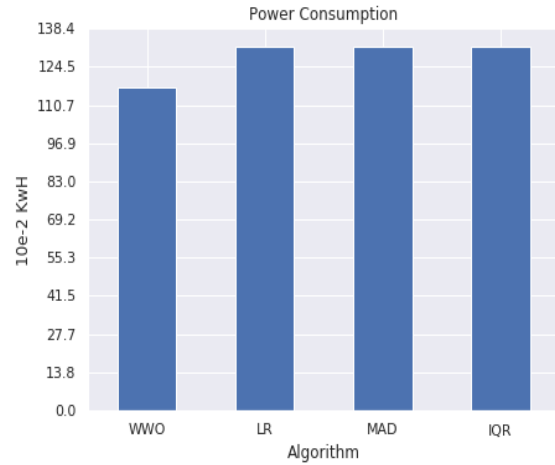


Fig. 7 120 Cloudlets

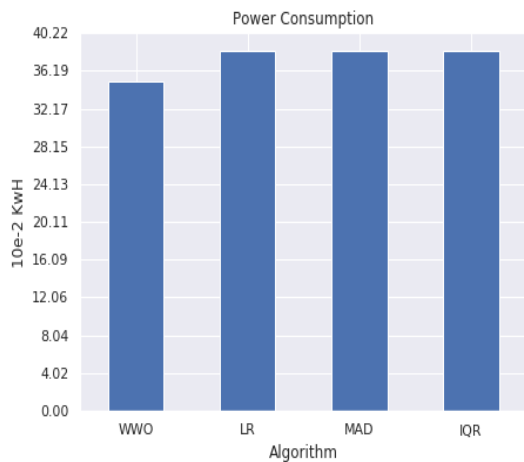


Fig. 5 60 Cloudlets

## VII. CONCLUSION AND FUTURE WORK

Cloud providers use different energy management strategies to maximize ROI. Energy-efficient VM consolidation is one such strategy to minimize monitoring costs of clouds. In this article, we present a dynamic efficient technique to place virtual machines on cloud servers by minimizing energy consumption. Our proposed WWO-based VM consolidation algorithm consumed 10 percent less energy compared to the standard dynamic migration algorithms like MAD, IQR, and LR. This algorithm has the scope of optimizations in the parts of modification of parameters and definition of operations of refraction, propagation, and breaking. Also runtime of the algorithm can be optimized so that the convergence of algorithm is faster.

The live VM migration is an expensive operation, so as a further effort, we plan to improve our proposed system model to the reduce number of VM migrations for energy conservation and maximizing system reliability.

## REFERENCES

- [1] Buyya, Rajkumar, et al. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." *Future Generation Computer Systems* 25.6 (2009): 599-616.
- [2] Rountree, Derrick, and Ileana Castrillo. *The basics of cloud computing: Understanding the fundamentals of cloud computing in theory and practice*. Newnes, 2013.
- [3] Goldberg, Robert P. "Survey of virtual machine research." *Computer* 7.6 (1974): 34-45.
- [4] Creasy, Robert J. "The origin of the VM/370 time-sharing system." *IBM Journal of Research and Development* 25.5 (1981): 483-490.
- [5] Uhlig, Rich, et al. "Intel virtualization technology." *Computer* 38.5 (2005): 48-56.
- [6] Zhu, Hai, and Hongfeng Wang. "New deadline-aware energy-consumption optimization model and genetic algorithm under cloud computing." *International Journal of Pattern Recognition and Artificial Intelligence* 30.03 (2016): 1659006.
- [7] Gupta, Rajeev K., and R. K. Pateriya. "Energy efficient virtual machine placement approach for balanced resource utilization in cloud environment." *International Journal of Cloud-Computing and Super-Computing* 2.1 (2015): 9-20.
- [8] Li, Kangkang, Huanyang Zheng, and Jie Wu. "Migration-based virtual machine placement in cloud systems." 2013 IEEE 2nd International Conference on Cloud Networking (CloudNet). IEEE, 2013.
- [9] Wu, Xiao-Bei, Jie Liao, and Zhi-Cheng Wang. "Water wave optimization for the traveling salesman problem." *International Conference on Intelligent Computing*. Springer, Cham, 2015.
- [10] Feller Eugen, Louis Rilling, and Christine Morin, "Energy-aware ant colony based workload placement in clouds", In Proceedings of the IEEE/ACM 12th International Conference on Grid Computing, IEEE Computer Society, pp.26-33, 2011
- [11] Liu Xiao-Fang, Zhi-Hui Zhan, Ke-Jing Du, and Wei-Neng Chen, "Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach", *ACM Proceedings of the Annual Conference on Genetic and Evolutionary Computation*, pp.41-48, 2014
- [12] Ferdaus Md Hasanul, Manzur Murshed, Rodrigo N. Calheiros, and Rajkumar Buyya, "Virtual machine consolidation in cloud data centers using ACO metaheuristic", In *European Conference on Parallel Processing*, pp.306-317, Springer, 2014
- [13] Wu Grant, Maolin Tang, Yu-Chu Tian, and Wei Li, "Energy-efficient virtual machine placement in data centers by genetic algorithm", In *Neural Information Processing*, Springer, pp.315-323, 2012
- [14] Ramezani, Fahimeh, Jie Lu, and Farookh Khadeer Hussain, "Task-based system load balancing in cloud computing using particle swarm optimization", *International journal of parallel programming*, Vol.42, No.5 pp.739-754, 2014
- [15] Wang Shangguang, Zhipiao Liu, Zibin Zheng, Qibo Sun, and Fangchun Yang, "Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers", *IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pp.102-109, 2013
- [16] Yassa Sonia, Rachid Chelouah, Hubert Kadima, and Bertrand Granado, "Multi-objective approach for energy-aware workflow scheduling in cloud computing environments", *The Scientific World Journal*, 2013.
- [17] Murtazaev, Aziz, and Sangyoon Oh. "Sercon: Server consolidation algorithm using live migration of virtual machines for green computing." *IETE Technical Review* 28.3 (2011): 212-231.
- [18] Craik, Alex DD. "The origins of water wave theory." *Annual review of fluid mechanics* 36 (2004).
- [19] Zheng, Yu-Jun. "Water wave optimization: a new nature-inspired metaheuristic." *Computers & Operations Research* 55 (2015): 1-11.
- [20] Mei, Chiang C., and P. L. Liu. "Surface waves and coastal dynamics." *Annual review of fluid mechanics* 25.1 (1993): 215-240.
- [21] Huang, Hung-Lung A. *Dynamics of surface waves in coastal waters*. Springer, 2010.
- [22] Zhao, Fuqing, et al. "A discrete water wave optimization algorithm for no-wait flow shop scheduling problem." *Expert Systems with Applications* 91 (2018): 347-363.
- [23] Farahnakian, Fahimeh, et al. "Using ant colony system to consolidate VMs for green cloud computing." *IEEE Transactions on Services Computing* 8.2 (2014): 187-198.
- [24] Calheiros, Rodrigo N., et al. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." *Software: Practice and experience* 41.1 (2011): 23-50.
- [25] Feitelson, Dror G., and Bill Nitzberg. "Job characteristics of a production parallel scientific workload on the NASA Ames iPSC/860." *workshop on job scheduling strategies for parallel processing*. Springer, Berlin, Heidelberg, 1995.
- [26] Cs.huji.ac.il. (2019). Parallel Workloads Archive: Standard Workload Format. [online] Available at: <http://www.cs.huji.ac.il/labs/parallel/workload/swf.html> [Accessed 3 October 2019].
- [27] Spec.org. (2019). SPECpower ssj 2008. [online] Available at: [http://www.spec.org/power\\_ssj2008/](http://www.spec.org/power_ssj2008/) [Accessed 3 October 2019]
- [28] Beloglazov, Anton, and Rajkumar Buyya. "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers." *Concurrency and Computation: Practice and Experience* 24.13 (2012): 1397-1420.