Comparative Evaluation of the Firefly, PSO and FCFS Scheduling Algorithms for Energy Management in Computer Data Center

E. O. Oyebode[1] and O. A. Ojesanmi[2+]
[1]Department of Computer Science,
Ajayi Crowther University,
Oyo,
Nigeria.
[2]Department of Computer Science,
Federal University of Agriculture,
Abeokuta,
Nigeria.
[+]Corresponding Email: dejioje@yahoo.com

ABSTRACT

The cloud service providers are often faced with the challenge of providing the computing resource in a manner that seeks to jointly meet the performance objectives of the application providers and users of the cloud. Energy cost on power consumption by host servers is one of the challenges in managing data center. To address this challenge, a Dynamic Resource Provisioning for Cloud Computing (DRP2C) for identifying cloud software resource requirements, load pattern and minimize energy resource usage was developed to enhance adequate utilization of cloud resources. The study examined dynamic resource provisioning using workload profiling approach. To achieve reduction in energy usage of data center, the energy usage of network devices was considered with the CPU utilization of server in the data center. The cloud resource allocation problem was formulated as two dimensional objective functions with a view to ensuring efficient energy and bandwidth utilization for resource optimization. The firefly algorithm was used in finding the optimum allocation for connecting virtual machine to the host server under various load scenarios and results obtained were evaluated. The simulation results showed that DRP2C has 86.8163kWh, 1.4528ms, 84.5000% while particle swarm optimization (PSO) has 116.2477kWh, 1.7591ms, 76.1000% and first come first served (FCFS) has 123.2400kWh, 1.9458ms, 66.7000%, for Energy Consumption (EC), Makespan (MS), Central Processing Unit Utilization (CU), respectively. Particle swarm optimization.

.

Keywords: Energy, utilization, profiling, server, provisioning

African Journal of Computing & ICT Reference Format

E. O. Oyebode and O. A. Ojesanmi. (2021), Comparative Evaluation of the Firefly, PSO and FCFS Scheduling Algorithms for Energy Management in Computer Data Center,
Afr. J. Comp. & ICT, Vol. 14, No. 2, pp. 29 –.39

1. INTRODUCTION

Energy efficiency is an important issue that relates to energy consumption and system performance in large scale computing systems such as cloud computing. A typical data centre has one or more compute servers and storage devices that are connected with network equipment that operates in a distributed fashion. It has been observed that the average energy consumption of servers will always be on the rise continuously [12]. With increase usage of servers, the energy consumption problem is of higher magnitude for large-scale infrastructures such as clusters, grids and clouds with heterogeneous devices. The type of operational costs maintained by the cloud service providers also vary depending on how cloud resource is controlled at runtime. Energy cost on power consumption, cooling cost and many more are examples of costs involved in managing cloud data centres on regular basis.

The cost of energy charges involve in maintaining the resource may vary with the number of compute servers and rate of utilizing such servers. CPU consumes greater part of energy in a typical server and many studies have focused on energy consumed by server by modeling the energy consumed via the CPU only. Meanwhile, the cost of the energy consumed through server for the period of its lifetime can out-weigh the original cost of procuring the server if not properly utilized. Other facilities that added to energy consumption in data center such as the cooling system and the power supply facilities etc may secure additional cost depending on the server utilization and policy for managing the data center. The facility cost and power supply cost is a function of the maximum utilization of the server [4].

To reduce unnecessary cost, it is essential to maximize the utilization of the servers and minimize other costs including communication costs. The cloud server machines when overloaded run the risk of high power consumption that leads to increase cost of maintenance. It has been shown also that underutilized servers consume more energy when compared to servers that are engaged in full use [7]. Lack of adequate use of cloud resources can further increase cost and eventually affect the efficiency of the service providers and other stakeholders in the cloud computing. An inefficient resource allocation policy can decrease the quality of services in the cloud system and cause service level agreement violation, increase cost of service and poor service performance [18].

Cloud computing requires that resource should be made available ahead of tasks in such a way that energy usage and other costs are reduced. Cloud service providers need to allocate resources to achieve optimum resource utilization that will make users enjoy their applications' performance requirements with minimum expenditure on power consumption [10]. Cloud resource provisioning efficiency can be determined using many important features such as periodicity, burstiness and the load-mix. These can be determined from the performance of each application of the cloud such as the different request types, amount of resources required, the tolerance of the application to provisioning decisions etc [11]. This study considered the additional communication cost of network devices with the CPU energy consumption in allocating virtual machine to the host in other to reduce energy usage. The study also considered allocating virtual machine to the host using the technique of firefly algorithm with the constraint of energy and communication costs

2. RELATED WORK

Swarm intelligence is one of the meta-heuristic techniques. It attempts to solve optimization problems by simulating the behavior of social insects. Swarm intelligence algorithms are used in complex distributed systems as it does not require any central control structure [1]. It offers flexibility in terms of addition and removal of resources. Metaheuristics approach has been applied to a wide variety of problems including resource management in cloud computing [15]. It can be used to offer multiple solutions to resource sharing in a dynamic environment such as found in cloud. The metaheuristic forms a high-level problem-independent method that can be used as a guiding strategy in designing underlying heuristics to solve specific optimization problems. The meta-heuristic scheduling has advantages over traditional techniques such as first come first served (FCFS), shortest job first (SJF), round robin (RR) etc. The traditional techniques are simple, fast and deterministic but do not adapt themselves with optimality problems [2]. Other techniques for sharing cloud resource involve the use of heuristic such as max-min, min-min, priority-based min-min, and enhanced max-min. The challenge with the heuristic technique is the issue of global minima and therefore do not guarantee to perform well in the cloud [16]. Swarm intelligence includes the use of genetic algorithm, particle swarm optimization (PSO), ant colony optimization, firefly etc.

Ferdaus [6] proposed scheme that can balance resource utilization of servers across different platforms with goal of minimizing power consumption and resource wastage. The study introduced the use of vector algebra to model multi-dimensional resource utilization using ant colony optimization (ACO). Resource parameters measured include CPU, Memory and Network I/O. The study introduced total resource capability as resource capacity vector, RCV while current resource utilization was represented as resource demand vector (RDV). The resource utilization of PM, p was summation of normalized resource utilization U_p^r of resource $r \in R$:

$$\text{Utilization}_p = \sum_{r\in R} U_p^r \quad (1)$$

and resource wastage was modeled as the summation of the remaining resources (normalized) using eq (1)

$$R = \{\text{CPU, Mem, Bd}\} \quad (2)$$

resource capacity vector (RCV), and resource utilization vector (RUV)

$$wastage_p = \sum_{r\in[R]}(1 - U_p^r) \quad (3)$$

Eq. (4) and eq. (5) is an expression for energy usage

$$U_p^{CPU} \in [0,1] \quad (4)$$

$$\text{with } E(p) = \begin{cases} E_{idle} + E_{full} - E_{idle}) * U_P^{CPU} & if\ U_P^{CPU} > 0 \\ 0 & otherwise \end{cases} \quad (5)$$

where E_{full} and E_{idle} represent the average energy drawn at busy and idle periods respectively.

Another approach used for resource allocation in the cloud included the use of fuzzy logic [9]. Fuzzy sets were applied to determine a cloud for delivering service and crisp sets were used to serve requests. Service availability was modelled as low, medium and high. Each of the fuzzy variables was represented with corresponding membership functions. Elmi [5] proposed a method for virtual machine placement. The approach used a 2-dimensional bin packing model that represented the physical machine as bin and the virtual machine as items to be packed with a view to minimize the number of physical machines involved and to balance the load of each machine. The formulation of the virtual machine (vm) considered central processing unit (CPU) and Memory as attributes of each virtual machine and how to minimize the energy consumption involved.

Pietri and Sakellariou [14] worked on strategies for mapping virtual machines (vms) unto permanent machines (pms). The work examined scheduling actions and how to optimize performance by considering various metrics. Mishra et al. [13] also study energy efficient virtual machine placement. The study carried out virtual machine configuration and placement and considered energy consumption and makespan with consideration for heterogeneous and dynamic tasks. Another related study on energy efficiency for cloud computing was carried out [17]. The study proposed an effective way to control energy usage in an infrastructure control cloud. The work adopted reinforcement learning style and fuzzy logic to effectively distribute resource allocation to manage revenue, cost and service level agreement (SLA) violation in a data center.

3. THE PROPOSED TECHNIQUE

The firefly algorithm is a stochastic optimization technique that considers the behavior of entity in order to find solution to a problem [8]. Each firefly is a representation of a solution with ability to affect the defined problem space using attractiveness [3]. At any given point in time, the brightness of the firefly in terms of the position is used to determine how good or how bad a solution is.

Each fly is represented with position and has brightness value. Each fly can be evaluated through the fitness function. The light intensity varies inversely with the square law as

$$I_{(r)} = \frac{I_s}{r^2} \quad (6)$$

Where I_s is the intensity at the source and considering the light absorption coefficient γ, the light intensity I varies with distance r such that

$$I = I_0 exp^{-\gamma r^2} \quad (7)$$

Similarly, the varied attractiveness β with change in distance can be defined as

$$\beta = \beta_0 exp^{-\gamma r^2} \quad (8)$$

β_0 represents the attractiveness at r = 0
Suppose m>0, γr^2 can be replaced by γr^m such that

$$\beta = \frac{\beta_0}{1+\gamma r^2} \quad (9)$$

The distance between any two fireflies can be computed using

$$r_{ij} = ||x_i - x_j|| \quad (10)$$

$$= \sqrt{\sum_{k=1}^{d}(x_{ik} - x_{jk})} \quad (11)$$

For two flies i and j to influence each other at time t, both having initial positions x_i
and x_j, the update may be obtained using:

$$x_i^{t+1} = x_i^t + \beta_0 exp^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha_t(\text{rand} - 0.5) \quad (12)$$

where t stands for the number of current iteration, α∈[0,1] represents the randomization parameter, rand is a small number less than 1 but greater than zero and γ=1.0 represents the coefficient of absorption for the firefly.

In the contest of applying the firefly algorithm to solve resource allocation problem, the virtual machine configurations are first obtained with respect to their energy usage and data transfer capacities and they are arranged such that they are all capable of handling tasks independently. The minimum data size of each task is also achieved ahead of the allocation by profiling application in AWS cloud.

In a datacenter Virtual machines have some major features for classifying them and hosting them in servers when needed,
Power consumed by servers can be computed using eq. (13) to eq. (20)

$$C_j = \begin{cases} C_j^{idle} + (C_j^{dynamic} - C_j^{idle}).u_j & if\ u_j>0 \\ 0 & otherwise \end{cases} \qquad (13)$$

u_j represents server utilization, C_j^{idle} represents idle power consumption of server, $C_j^{dynamic}$ and represents dynamic power consumption.
For switches involved, energy usage can be obtained using

$$C_k^{switch} = \begin{cases} C_k^{static} + C_k^{port}.n_k & if\ switch\ k\ is\ on \\ 0 & otherwise \end{cases} \qquad (14)$$

n_k = the number of active ports on switch k

$$\text{Minimize } \sum_{i=1}^{M} \boldsymbol{C_i^{server}} + \boldsymbol{C_i^{switch}} + \sum_{n=1}^{N} \boldsymbol{CT_{ij}^{n}} \ \ \forall \boldsymbol{i} \in \boldsymbol{M}, \forall \boldsymbol{j} \in \boldsymbol{N} \qquad (15)$$

where CT refers to the data traffic between server i and server j subject to the constraints

$$\sum_{j=1}^{n} X_{ij} = 1, \forall i \in Y \qquad (16)$$

Y represents the division of cloud resource to different instance types inform of virtual machine, vm

Equation (14) ensures that each running instance can only be assigned to one instance type

$$\sum_{i=1}^{n} X_{ij}.P_i \le P_j' \qquad (17)$$

P_i represents total memory allocation on virtual machines meant for server j
P_j' represents total memory allocation on server
Equation (15) ensures that each memory allocation do not exceed the total server capacity memory

$$\sum_{j=1}^{n} X_{ij}.R_i \le R_j' \qquad (18)$$

Equation (16) ensures that each million instruction per seconds (mips) allocated do not exceed the total Mips of server capacity

$$\sum_{j=1}^{n} X_{ij}.C_i \le C_j' \qquad (19)$$

Equation (17) ensures that each bandwidth allocated do not exceed the total server bandwidth capacity

memory utilization, $$Mem_u = \frac{m_p}{v_r}, p \in P, r \in R \qquad (20)$$

m_p refers to the memory requirement of the profile application
v_r refers to the memory released by the virtual machine

Mathematical calculations and comparisons can be resolved quickly and effectively with MATLAB and that is why the proposed Firefly algorithm was implemented in MATLAB. The problem has been formulated as a linear programming problem solvable in MATLAB 2016.

This study has implemented the firefly algorithm in eq (7) – eq (10) in the same environment alongside with other methods. The study represented power usage of server as real number using eq (11); the objective function stated in eq (13) was associated with the light intensity of the fireflies. The parameters of the firefly were set, with the input parameters as shown in Algorithm1 and the output of the algorithm was used as solution for the allocation problem.

The firefly algorithm used in the simulation is shown in Algorithm 1.

Algorithm 1: Resource provisioning algorithm

Input :vmlist:v_m Task:ts_t Ffiy:b max_iter:m, serv_list:s
Output: alloc(server,vm)

```
begin
    count ← 1
    While (count < max_iter)
        for t ∈ ts
            for v ∈ vm
                Q ← assignrandom(t,v)
            end for
        end for
    end while
    n← 1
    for p ∈ ts
        y_p←assgn(p,Q)
        Fs ← evaluate(y_p) using equation (6) – (9)
    end
    gmin ← computemin(Fs)
    for i ← 1: |Fs|
    if Fs(i) >Fs(i+1)
        let k ∈ (Ts(i), Ts(i+1))
        new_Fs(i) ← reassign (Ts(i), (Ts(i+1),k)
    end
    update Fs using equation (10)
    count ← count + 1
    Final_gmin ← min(gmin)
    end
    end begin
```

4. PERFORMANCE EVALUATION

Two services were developed and hosted in amazon web services (AWS) and the Cloud watch of the amazon web services (AWS) was used to carry out monitoring of cloud resource usage. The developed services were used in profiling the patterns of the resource usage on the AWS cloud. The captured features were used in MATLAB for setting up the simulation environment for allocating virtual machines to host servers using proposed provisioning technique along side with particle swarm optimization (PSO) and first come first served.
Figure 1 shows the plot of normalized dataset for bandwidth, CPU and memory normalization obtained when the services were hosted in Amazon Web Services and monitored via the cloud watch.

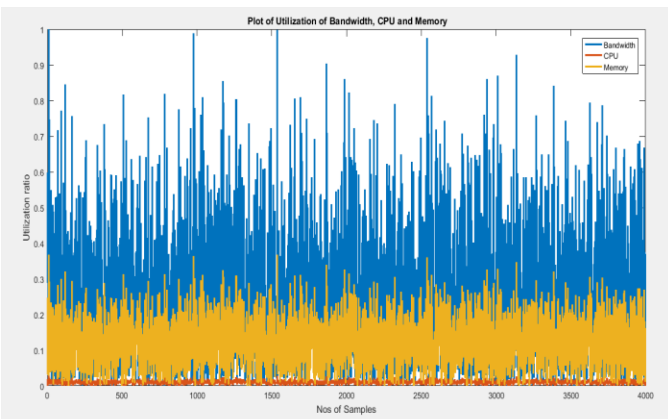

Fig. 1 Plot of normalized dataset for bandwidth, CPU and memory normalization.

Table 1 contains the details of the parameters used for the configuration of the host machines in the simulation. Table 2 also contains the summary of the virtual machine configuration setup for the purpose of the simulation. Table 3 contains the description of the properties used for firefly algorithm, particle swarm optimization and first come first served algorithms that were used in carrying out the allocation of virtual machines to the host. Cloud computing resource performance in the simulation environment was evaluated using important criteria that can reflect the impact of the energy usage on the proposed DRP2C that used the firefly technique, PSO and FCFS in allocating resources. The performance evaluation criteria used are: Energy consumption rate (ECR), Makespan and CPU utilization (CU).

Table 1: Summary table for parameters used for servers in simulation

Host type	Number of Cores	Bandwidth (Mb/s)	RAM (Gb)	CPU (Mips)
High CPU	4	80000	10	2400
Extra	4	40000	6	1600
Small	2	20000	4	800
Micro	2	10000	2	800

Table 2: Summary table for parameters used for virtual machines in simulation

VM Type	Bandwidth (Mb/s)	RAM	CPU MIPS
Xlarge	1600	8096	800
Large	800	8096	800
Medium	400	4048	400
Small	200	1024	200

Table 3: Summary table for parameters used in DRP2C, PSO and FCFS

Simulation parameters	Value
Mutation Coefficient	0.2
Number of Fireflies (Swarm Size)	25
Light Absorption Coefficient	1
Number of iteration	100
Attraction Coefficient Base Value	2
Mutation operator	0.2
Learning factor	1.5

Energy consumption rate (ECR): This was used to measure the rate of energy usage among the FCSF, PSO and DRP2C models when virtual machines are connected to the host and resource required. Figure 1 shows the energy consumption of the first come first served algorithm, the PSO and the DRP2C as each of the model was used in recording the rate of energy usage. Between the first three hours, the rate of energy consumption did not show significant difference but as more requests were generated the changes continue to increase. After engaging the models beyond simulation time of 7hrs, the rate of energy consumption manifested by the three models became noticeable. While the FCFS model exhibited highest energy consumption, it was followed by PSO which also used optimization in allocating virtual machines to the server. DRP2C also had increase in the power consumption at a lowest rate when compared to others.

Table 4: Summary Table for energy consumption rate

Time (Hr)	Energy Consumption Rate (kWh)		
	FCFS	PSO	DRP2C
1	0.3241	0.5941	0.4202
2	0.8934	1.0620	0.9005
3	1.6850	2.2940	1.7080
4	5.2740	5.1240	3.7680
5	6.5430	8.7590	6.5740
6	10.5500	13.8100	10.5800
7	17.8800	22.3500	15.9000
8	24.8800	27.0900	20.5800
9	35.7700	33.6100	26.9000
10	54.1200	53.7800	36.2900
11	53.6700	49.8700	39.7400
12	72.9800	68.7300	49.8100
13	86.9800	83.1800	61.5200
14	94.8300	92.9700	70.4900
15	110.8000	107.0000	82.9500
16	146.9000	133.8000	98.4800
17	156.4000	149.7000	111.3000
18	179.7000	165.7000	123.4000
19	185.4000	178.9000	137.4000
20	225.7000	213.1000	157.2000
21	246.5000	232.6000	172.4000
22	301.9000	280.8000	207.7000
23	322.6000	303.9000	226.8000
24	360.2000	321.0000	244.5000
25	378.8000	356.5000	263.0000

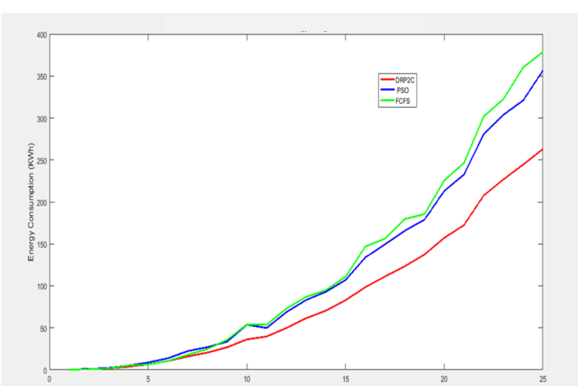

Fig. 2: Plot of energy consumption for DRP2C, PSO and FCFS

Makespan: The makespan was measured as the total time used by the cloud resource to complete the execution of all tasks. The completion time of the last tasks in the vms were used as the makespan. The readings taken for Makespan are shown in Table 5. The makespan of the three approaches is as shown in the Figure 2. As the number of tasks increases, the makespan of DRP2C became smaller than that of PSO and FCFS. The highest makespan obtained for the DRP2C is 1.892000ms while that of the PSO is 2.125000ms while that of the FCFS reached 2.657000ms and it occurred to all the three models when the number of virtual machines exceeded 100. When the number of virtual machine increased to 200, the FCFS reached the highest makespan attained which is 2.657000ms and it occurred to all the three models when the number of virtual machines in use exceeded 100.

Table 5: Summary Table for Makespan

Number of vms	Makespan (ms)		
	FCFS	PSO	DRP2C
10	0.5712	0.5249	0.5423
20	0.7996	0.7689	0.7588
30	1.1240	0.9590	1.0130
40	1.5290	1.3360	1.1240
50	1.7520	1.5310	1.3150
60	1.8140	1.5610	1.3900
70	1.8120	1.6550	1.2890
80	1.9590	1.7460	1.4880
90	1.9980	1.6720	1.5990
100	1.9050	1.7350	1.2850
110	2.1740	1.9820	1.7330
120	2.1200	1.9930	1.5360
130	2.0600	2.0060	1.5230
140	2.1250	2.0020	1.6110
150	2.1060	2.0110	1.6330
160	2.0470	2.0130	1.5640
170	2.0710	2.0170	1.5390
180	2.1160	2.0200	1.5750
190	2.1380	2.0340	1.6420
200	2.1890	2.0620	1.5410
210	2.2160	2.0780	1.5680
220	2.3150	2.0890	1.6190
230	2.4570	2.1250	1.7180
240	2.5890	2.0240	1.8230
250	2.6570	2.0330	1.8920

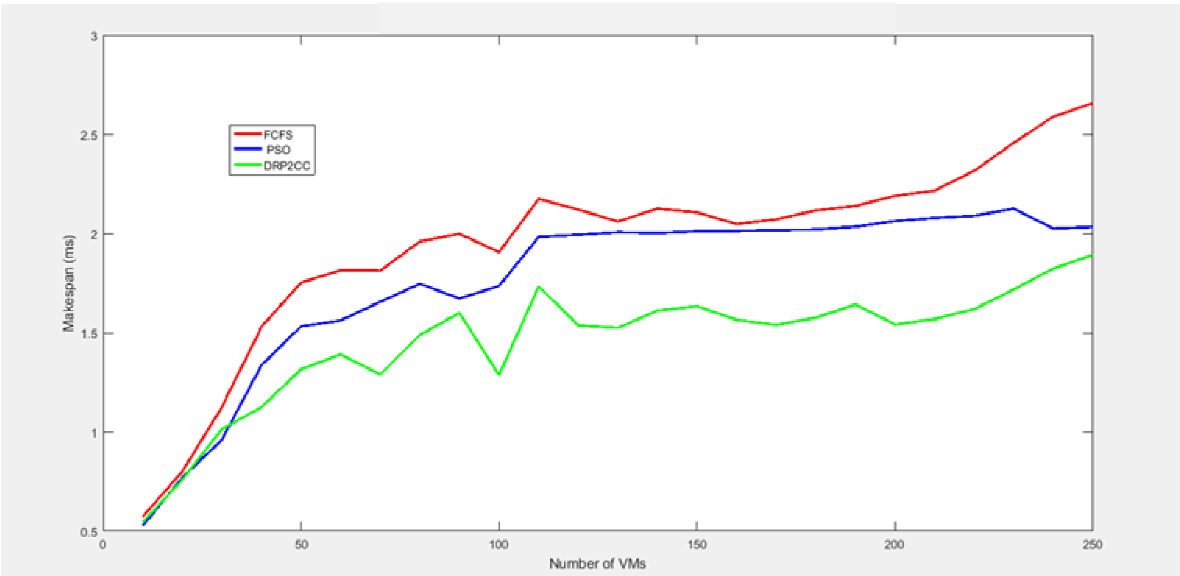

Fig. 3 Plot of makespan for DRP2C, PSO and FCFS.

CPU utilization (CU): The CU shows the rate at which the servers in a data centre is engaged. It is a measure that is related to the power consumption rate in a data centre. The Summary Table for CPU utilization % is shown in Table 6.

(a) The CPU utilizations are as shown in Figure 3. When the number of vm used in the simulation is between 10 - 80, the average utilization of PSO was nearly the same as that of DRP2C but as the number of vm increased to 120, the CPU utilization of DRP2C became higher and showed clear difference among the three models. The CPU utilization of FCFS reached a peak level of 66.7000% while that of PSO reached 76.1000% and that of DRP2C reached 84.5000%.

Table 6: Summary Table for CPU utilization %

Number of vms	CPU utilization %		
	FCFS	PSO	DRP2C
10	45.0000	53.0000	54.0000
20	47.3000	51.0000	57.0000
30	48.8000	59.2000	58.9000
40	49.7000	59.4000	59.3000
50	49.9000	61.5000	62.1000
60	50.2000	63.1000	63.2000
70	56.7000	62.3000	63.5000
80	57.3000	60.5500	63.6000
90	58.8000	59.8000	63.8000
100	59.1000	60.4000	65.3000
110	59.5000	63.0000	68.0000
120	59.7000	64.5000	70.1000
130	60.2000	66.7000	72.3000
140	61.1000	68.5000	73.8000
150	62.3000	68.8000	74.1000
160	63.7000	69.3000	74.5000
170	63.9000	70.0000	75.7000
180	64.0000	71.1000	76.2000
190	64.2000	72.1000	76.4000
200	65.7000	73.3000	77.3000
210	66.7000	73.7000	78.5000
220	66.3000	74.5000	79.5000
230	65.0000	74.9000	80.2000
240	65.3000	75.2000	83.2000
250	65.4000	76.1000	84.5000

performance characteristics of the cloud. The study used cloud watch of AWS to measure resource usage. Using the knowledge gained during analysis, the study used MATLAB simulation for representing shared resources and execution of tasks in the cloud via vm. DRP2C used computing techniques of firefly to search and discover in optimization the best combination of vms and host servers that could lead to improve efficiency in terms of energy cost. The performance evaluation was carried out using energy consumption rate, makespan and CPU utilization percentage.

5. CONCLUSION

Focusing on data center energy usage, this study aims at understanding, modelling and improving cloud resource performance provisioning in the cloud. The study considered minimization of core resources available in mapping virtual machines to host servers. The study used MATLAB simulation for representing shared resources and execution of tasks in the cloud via virtual machine (vm). DRP2C used computing techniques of firefly to search and discover through optimization, the best combination of virtual machines (vms) and host servers that could lead to improve efficiency in terms of energy cost. The performance evaluation was carried out using energy consumption rate, makespan and central processing unit (CPU) utilization percentage.

REFERENCES

[1] R. A. Al-Arasi and A. Saif. Task scheduling in cloud computing based on metaheuristic techniques: A review paper. EAI Endorsed Transactions on Cloud Systems 6(17):1-19, 2020.

[2] N. Arora and R. K. Banyal. HPSOGWO: A Hybrid Algorithm for Scientific Workflow Scheduling in Cloud Computing. International Journal of Advanced Computer Science and Applications 11(10):626-635, 2020.

[3] P. Chahal and V. Saroha. Energy Efficient Resource Allocation by Firefly Optimization in Cloud Computing. International Journal of Recent Research Aspects 3(3):20-24, 2016.

[4] S. Diouani and H. Medrom. An Adaptive Autonomic Framework for Optimizing Energy Consumption in the Cloud Data Centers. International Journal of Intelligent Engineering and Systems 12(4):111-121, 2019.

[5] S. M. Elmi. Solve the Packing Problem of Virtual Machine Placement based on PSO algorithm. International Journal of Modern Engineering Research 7(12):42-56, 2017.

[6] H. Ferdaus, M. Murshed, R. N. Calheiros and R. Buyya. Virtual Machine Consolidation in Cloud Data Centers Using ACO Metaheuristic. 20th European International Conference on Parallel Processing 306-314, 2014.

[7] Y. Saadi and S. El Kafhali. Energy-efficient strategy for virtual machine consolidation in cloud environment. Soft Computing 24(2):14845–14859, 2020.

[8] I. A. Saleh, O. I. Alsaif, S. A. Muhamed and E. I. Essa. Task Scheduling for cloud computing Based on Firefly Algorithm. Journal of Physics: Conference Series pp 1-7, 2019.

[9] A. A. Sharieh and L. Albdour. A Heuristic Approach for Service Allocation in Cloud Computing. International Journal of Cloud Applications and Computing 7(4):60-74, 2017.

[10]S. Singh, I. Chana, and R. Buyya. STAR: SLA-aware Autonomic Management of Cloud Resources, IEEE Transactions on Cloud Computing (TCC) 8(4): 1040-1053, 2020.

[11] P. Ilia and S. Rizos. Mapping Virtual Machines onto Physical Machines in Cloud Computing: A Survey. ACM Computing Survey 49(3):1-30, 2016.

[12] L. Ismaila and A. Fardoun. EATS: Energy-Aware Tasks Scheduling in Cloud Computing Systems. The 6th International Conference on Sustainable Energy Information Technology. Pg 870 – 877, 2016.

[13] S. K. Mishraa, D. Puthal, B. Sahooa, P. P. Jayaramanc, S. Jun, A. Y. Zomaya. and R. Rajiv Ranjan. Energy-efficient VM-placement in cloud data center. Sustainable Computing: Informatics and Systems 20:48-55.

[14] L. Pietri and R. Sakellariou. Mapping Virtual Machines onto Physical Machines in Cloud Computing: A Survey. ACM Computing Surveys, 49(3), 2016.

[15] G. Rjoub and J. Bentahar. Cloud Task Scheduling based on Swarm Intelligence and Machine Learning. IEEE 5th International Conference on Future Internet of Things and Cloud 272 – 279, 2017.

[16] T. Thandar, M. M. Myat, P. Sazia. and G. Amjad. Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers. Journal of King Saud University - Computer and Information Sciences, 2018. doi.org/10.1016/j.jksuci.2018.11.005

[17] T. Thein, M. M. Myo, S. Parvin. and A. Gawanmeh. Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers. Journal of King Saud University – Computer and Information Sciences, 2018. https://doi.org/10.1016/j.jksuci.2018.11.005.

[18] Z. Zhan, X. Liu, Y. Gong, and J. Zhang. Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches. ACM Computing Surveys, Vol. 47(4):1-33, 2015.