A Novel Approach of Reducing Energy Consumption by Utilizing Enthalpy in Mobile Cloud Computing

Mostafa Abdulghafoor MOHAMMED*, Nicolae ȚĂPUȘ
Faculty of Automatic Control and Computers,
University Politechnica of Bucharest
313 Splaiul Independenței, 060042, Romania
*(*Corresponding author)* alqaisy86@gmail.com

Abstract: The Mobile Cloud Computing (MCC) technology is a growing technology that aids in improving the quality of mobile services. The resources in MCC are dynamically allocated to the users based on their needs. The users pay for the resources consumed by their programs, but the drawbacks of process failures and knapsack problems of resource allocation still exist in MCC. Furthermore, the scheduling of energy consumption and computational cost is very high. To solve these issues, an optimized energy efficiency resource management technique is set forth in this study. The recommended method holds two stages: a) the initial stage, when the task loss, transmission probability, delay, utilization and reputation for every single task is individually measured and the enthalpy was calculated, and b) the second stage, when the enthalpy-related Cuckoo Search Optimization (CSO) algorithm was used to optimize and prioritize the resources to the powerful resource management. The execution of the recommended method automatically reduced the knapsack issue, energy and cost. A formal analysis of the resource management framework has ensured that the provider executes resources based on the power consumption. The performance of the suggested algorithm was benchmarked against the performance of other conventional algorithms.

Keywords: Task measures, Optimization, Enthalpy, Resource management, Task prioritization.

1. Introduction

The MCC is a promising technology that introduced cloud computing into mobile computing; where mobile devices are connected to the Internet via a wireless network and communicate with the remote cloud seamlessly [1]. The combination of MCC with cloud computing proffers a cloud computing environment through different mobile devices [1],[2]. The MCC technology is mainly comprised of three components, which are mobile internet, mobile computing, and cloud computing [3][4]. Cloud computing is notably becoming the primary computing service delivery platform, while mobile computing is becoming the end-user device of selection, specifically for sustaining integrated exterior services [5][27].

Nowadays, the main researches in the area of MCC are based on the deduction of the mobile device energy consumption, coupled with making more offloading decisions [6]. The mobile cloud networking is a mushrooming technology whereby mobile devices are connected to a cloud server. The cloud server distributes network resources such as available bandwidth and stores the clients based on demand so that they can perform optimally in real time without any performance degradation [7]. With the rapid advances being made in mobile computing, mobile devices are now able to perform a broader variety of tasks as matched with what they can do a decade ago [22][24]. In fact, several tasks such as video calling and game playing, which were mainly computer accomplished earlier can now be performed on the mobile devices. However, powerful applications necessitate the need for high computational power owing to their data processing requirements and involvement of other computationally intensive operations [8][28]. The introduction of MCC is a possible solution to the inherited restrictions of mobile computing [9][29].

The MCC was introduced as an efficient system for the resource-intensive and latency-sensitive mobile applications [2]. Furthermore, cloud computing has been the function of cloud resources which are sent like a service through the network [10][11]. Cloud computing has become the platform for the selection several applications owing to the benefits of high calculation power, high scalability, low service cost, accessibility, and availability. Cloud computing is an important part of different fields such as commerce [12], transportation [13], social networks [14] health care, and education [15].

In the recent era, the advancements in computation and communication processes have increased the use of mobile devices, and this has led to the

https://doi.org/10.24846/v26i4y201706

dissipation of a huge amount of energy [16]. The management of the network and cloud resources in cloud computing comes with several challenges, and to address these challenges, it is necessary to develop a resource management approach which ensures the scalability, manageability, adaptability, efficiency, and dependability of cloud computing [25][15]. The management of MCC resources is also a great challenge owing to the number of objectives that need to be met while considering the relative size (large) of the infrastructures [17][20]. Few studies have previously addressed these issues [18] [19] [20][26], and as a consequence, it is a salient issue for service providers to design powerful resource management schemes that will satisfy the requirements of the MCC applications in real time[21].

This paper is outlined as follows: Section 2 reviewed the related works regarding the recommended method, while Sections 3 provided a brief discussion of the suggested methodology. Section 4 analyzed the evaluation results from the several experiments conducted in this study, while Section 5 concluded the paper.

2. Related Works

A systemic framework for the MCC hierarchical resource sharing system proposed by Awais Ahmad *et al.* [2] was categorized into three domains, which are Local ISP Server (LIS), Global Cloud Server (GCS), and Gateway Server (GWS). Similarly, a new system for the alleviation of network delay based on the deployment of foglets at each of the proposed clustering algorithm has been presented. The Fuzzy rule-based algorithm was proposed for the elimination of unrequired foglets prior to the determination of the optimal foglet for handover. A technique for foglet selection has been enhanced based on the TOPSIS decision mechanism. The divergent parameters (including delay, Bit Error Rate, jitter), communication cost, response time, packet loss, and network load were considered before choosing an optimum network.

The aim of the resource allotment was to select service providers and reduce the offloading time to optimize the service life of mobile devices and fulfilling their deadline constraint.

Ghasemi-Falavarjani *et al.* [23] have proposed a two-stage method towards rectifying the issue: initially, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) was used to derive the Pareto solution set; next, entropy weight and TOPSIS were used to determine the best compromise solution. Moreover, a context-aware offloading middleware has been enhanced for collecting contextual information and handling offloading procedures.

Yanchen Liu *et al.* [15] had suggested an energy-effective arrangement of works, where mobile devices transfer packets to the cloud via a Wi-Fi access point. The scheduling system aims to reduce the rate of energy consumption in mobile devices under the constraint of total completion time for one application. The work scheduling issue was rebuilt into a controlled smallest path, while the LARAC technique was used for receiving the estimated optimum solution. The suggested energy-effective strategy decreases energy consumption. Moreover, the effectiveness and performance of the proposed system were ascertained in several applications, where there was a difference between the time constraint and the workload ratio between communication and computation.

A double-sided bidding system in which the bids for the selection of a demand resource-price function are provided by the demanding users while the supplying user provides the bid for the selection of the supply resource-price function has been proposed and analyzed by Ling Tang *et al.* [25]. The case was considered along with users (price-taking and price-anticipating) who would forecast the impact of their own bids on the price. The price-taking users believe that the suggested mechanism initially allows a unique competitive equilibrium which augments the mobile clouds' social wellbeing, and then, develops an optimum distributed algorithm for achieving the intended equilibrium point. The price-anticipating users, on the other hand, will initially assemble the interaction between several users and confirm the uniqueness and existence of Nash equilibrium; and next, enhance a distributed algorithm for calculating the Nash equilibrium.

Rakpong Kaewpuang *et al.* [11] proposed the formation of a resource assignment for mobile applications and the establishment of revenue management scheme and cooperation formation among service providers. For the allocation of network resources to mobile applications, they developed and optimized the models for the achievement of the optimum number of applications which can be supported for the optimization of the revenue of the service providers while facing the requisition of the mobile applications. For the distribution of the generated revenue among the mobile cloud service providers, they applied the core plus Shapley value concept based on the cooperative game theory as a solution. Regarding the revenue shares, the cloud service providers can decide how to cooperate and allocate their resources to the resource pool. The provider can also decide a number of resources to commit to the resource pool.

3. Proposed Methodology

This paper sets forth a powerful optimized energy resource management scheme for mobile cloud computing. The suggested scheme is described in two stages; first, various task measures were taken and based on the task measurement, the enthalpy was calculated. In the second stage, the enthalpy-related CSO algorithm was engaged for the powerful resource management scheme. Here, the optimization algorithm pre-arranged the tasks based on the optimum values for the assigned resources in the mobile cloud. The maximized enthalpy values were used for receiving the scheduled tasks based on the priority. The block diagram of the recommended approach is presented in Figure 1.

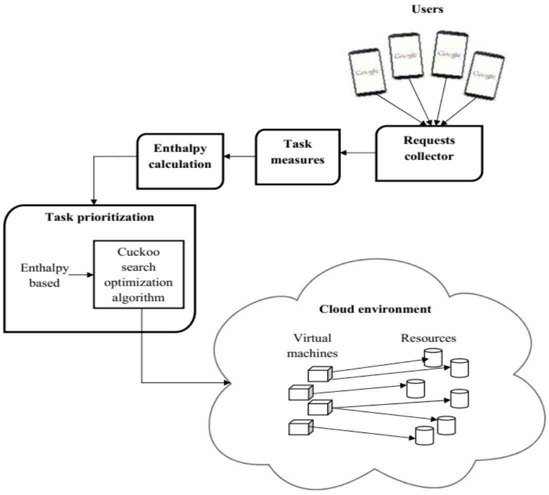


Figure 1. Block diagram of the proposed method

Powerful resource management

In a dynamic cloud environment, different requests are submitted by different users, and each request is made up of different tasks. In a cloud computing environment, the server is the service provider that will handle the tasks, and for every task, different measures such as task loss, reputation, delay time, transmission probability, energy utilization, and enthalpy are computed for achieving a better resource management. Here, the cloud administrator has a significant role in the proposed powerful resource management scheme because it decides the priority among the various users.

3.1 Task measures calculation

In resource allocation, congestion may result from assigning many tasks to the task collector at the same time. To avoid this situation, a powerful resource management is important. In this paper, divergent task measures such as task loss, delay time, reputation, transmission probability, energy utilization and enthalpy were measured as described below.

3.1.1 Task loss

The task loss was measured by the number of works available in the schedule and the number of tasks that were scheduled correctly using Equation (1),

$$\mathrm{T_L} = \frac{T(n) - T(m)}{T(n)}, \tag{1}$$

where $T(n)$ is the total task available before scheduling, $T(m)$ is the total tasks that were scheduled, while $\mathrm{T_L}$ is the task loss.

3.1.2 Delay time

Packets that travel through a data center may disappear before reaching their destination due to link errors. These losses may cause salient communication delays. A delay is described as the standard time consumed to complete a task. The delay is obtained by considering the difference between the time of sending the packet and the time of its receipt. Assume the additional time taken by the task to end as latency $\mathrm{L_t}$ and $\mathrm{T_r}$ be the time of the transfer, the delay time D_t is calculated using Equation 2, while the transfer time is calculated using Equation 3 and the latency, by Equation 5.

$$D_t = \mathrm{L_t} + \mathrm{T_r} \tag{2}$$

$$\mathrm{T_r} = \frac{D}{B_w} \tag{3}$$

where D denotes the single task time and B_w is the bandwidth per user. Here, B_w is measured by Equation 4,

$$B_w = \frac{B_{tot}}{N}, \tag{4}$$

where B_{tot} is the total available bandwidth, and N is the total number of requests. Latency is the total task waiting time calculated as:

$$\mathrm{L_t} = T_{wait}, \tag{5}$$

where $\mathrm{L_t}$ is the latency, T_{wait} is the total waiting time of long tasks.

3.1.3 Reputation

This is a measure of the service trust in the context of a resource provider. The reputation ranking is given by the client as:

$$R_v(r_q) = \tau_q \times V_q \sum_{i=1}^{N} (b_q + (U_q \times a_q) \tag{6}$$

$$V_q = \exp(\mu - \rho_q) \tag{7}$$

The award factor V_q describes the trust dynamics on historical behaviors, μ is the received task, and τ_q is the delay factor.

$$\rho_q = \frac{\mu_q}{N_q}, \tag{8}$$

where N_q is the number of executed tasks, μ_q is the number of received tasks, b_q is the VM index, U_q is the node count, a_q is the access point index.

3.1.4 Transmission probability

The possibility of successful transmissions over the network is referred to as the transmission probability, calculated as:

$$T_{prob} = \frac{1}{\gamma(s,a)}, \tag{9}$$

where γ is the discount factor.

3.1.5 Utilization

This is the number of user cycles (%) absorbed by the CPU cores on the VM nodes in the IaaS cloud which are either performing a computing task or an on-demand user VM. Utilization is measured using Equation 10.

$$\mathrm{U} = \frac{\sum_{\mathrm{j}} \mathrm{V_{ms}\,CT_j}}{\mathrm{S_m \times V_m}} \tag{10}$$

In Equation 10, S_m denotes the 'make span', which is the time used to complete the task, V_{ms} denotes the available virtual machines, C is the number of failures, T_j denotes the tasks, and V_m is the total virtual machine count.

3.1.6 Energy

The key aim of computing the energy consumption of a node is to avoid delays in the node's performance, save energy, and reduce cost. When the nodes are idle, power is being released and the remaining energy for the packets and the calculations denotes the power spent in the computations that occur within the routing nodes and power tunings. The energy consumed by a node within time t is estimated using Equation 11,

$$E_{c(t)} = N_T * A + N_R * B, \tag{11}$$

where $E_{c(t)}$ represents the absorbed energy by the node after time t, N_T and N_R represents the number of transmitted and received packets respectively by the node, A and B are constant factors based on the energy model.

3.2 Enthalpy measure

Enthalpy is denoted as a state function which relies only on the existing equilibrium state determined through the internal energy of the system. Here, the enthalpy was computed using the above-computed task measures. A system's enthalpy is designated as:

$$\mathrm{H} = \mathrm{R_v(r_q) + PV} \tag{12}$$

$$\mathrm{PV} = (\mathrm{T_L \times D_t \times U \times T_{prob}}), \tag{13}$$

where H is the enthalpy of the system, $R_v(r_q)$ is the reputation, D_t is the delay time, $\mathrm{T_L}$ is the task loss, U is the utilization, and T_{prob} is the transmission probability. The suggested CSO algorithm uses such enthalpy values in maximization.

3.3 Enthalpy based cuckoo search optimization

The tasks were prioritized using the enthalpy-related CSO algorithm. The calculated enthalpy values were used as the input to the CSO algorithm. The suggested CSO algorithm helps the cloud admin to decide the user priorities and allocate resources efficiently based on the priorities.

3.3.1 Cuckoo Search optimization algorithm

The CSO is a heuristic search algorithm recently suggested by Yang and Deb. The algorithm was modelled after the reproduction policy of cuckoos. The cuckoos lay their eggs at the fundamental level in the nest of other host birds (maybe same specie or different species). The eggs may be found by the host bird who recognizes it as not its own and may decide either to destroy the egg or abandon the nest together. Here, the CSO algorithm uses the computed enthalpy (H) values in maximization. Figure 2 showed the pseudo code of the proposed CSO.

```
begin
Objective function $f(y)$, $(y_1, y_2, y_3, .... y_p)$
Generate the initial population of $n$ host nests
$(Y_i = 1,2,3,....n)$
while ($t$<MaxGeneration) or (stop criterion)
Get a cuckoo randomly by Levy flights
evaluate its quality/fitness $F_j$
Choose a nest among $n$ (say, $i$) randomly
if ($F_i < F_j$),
replace $j$ by the new solution;
end
A fraction ($p_a$) of worse nests are abandoned and new ones are built;
Keep the best solutions
(or nests with quality solutions);
Rank the solutions and find the current best
end while
Postprocess results and visualization
end
```

Figure 2. Pseudo code of CSO algorithm

Algorithm description/procedures

The CSO algorithm improves the dependability and sturdiness in handling optimization issues. The steps of this weight optimization algorithm are illustrated below.

- ***Initialization***

Suggest a random population of n host nests Y_i $(Y_i = 1,2,3,....n)$.

- ***Levy Flight Behavior***

Derive a cuckoo by Levy flight behavior equation that is signified as follows,

$$\mathrm{Y_i}(\mathrm{t}+1) = \mathrm{Y_i}(\mathrm{t}) + \alpha \oplus \mathrm{Levy}(\lambda), \alpha > 0 \quad (14)$$

$$\mathrm{Levy}(\lambda) = \mathrm{t}(-\lambda), 1 < \lambda < 3 \quad (15)$$

- ***Fitness Calculation***

Calculate the fitness using the fitness function (F_i) for obtaining an optimum solution. Choose a random nest j. While the fitness function's value of the cuckoo egg is lesser than or else equivalent to the fitness function value of the haphazardly chosen nest (that means $F_i < F_j$), then, the haphazardly selected nest (j) is replaced by the new solution. Here, the enthalpy (H) values calculated in (12) are used in the fitness function to obtain the optimal calculation.

$$Fitness\ function = C_{best} - P_{best}, \quad (16)$$

where C_{best} is the current best solution, and P_{best} is the current best solution. A fitness function value approaching the value zero denotes that the difference between the solutions has reduced due to the augment in the number of repetitions.

The fitness is the differentiation in solutions and the innovative solution is substituted by the haphazardly selected nest; or else, while the cuckoo egg's fitness is more than the randomly selected nest, the host bird considers the alien egg as non-self and may result in tossing the egg or abandoning the nest.

- ***Termination***

In the recent iteration, the solution is matched and the most excellent solution is only passed further, which is done by the fitness function. If the number of iterations is less than the optimum, then, it maintains the best nest. The CSO will terminate after maximum iterations have been reached and the optimal solution has been achieved. The flow diagram of the CSO algorithm is presented in Figure 3.

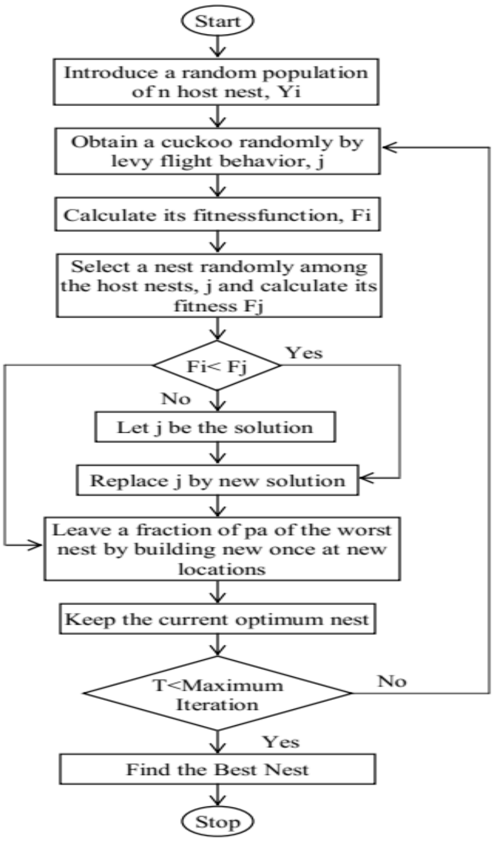

Start
Introduce a random population of n host nest, Yi
Obtain a cuckoo randomly by levy flight behavior, j
Calculate its fitnessfunction, Fi
Select a nest randomly among the host nests, j and calculate its fitness Fj
Fi< Fj
Yes
No
Let j be the solution
Replace j by new solution
Leave a fraction of pa of the worst nest by building new once at new locations
Keep the current optimum nest
T<Maximum Iteration
No
Yes
Find the Best Nest
Stop

Figure 3. Flow diagram of CSO

In the proposed algorithm, the resources are assigned based on the task priority level. This priority-based resource assignment resolved the problems in resource allocation and ensured an efficient resource management.

Resource management is important in sharing computing resources between clients. The efficient management of resources in data centers has a role to play for both customer satisfaction and profit maximization. The proposed enthalpy-related CSO algorithm assists the cloud service admin when deciding the user priority level for efficient resource allocation based on the priority. This resource allocation method is more effective than the prevailing ones, since, in those systems, no priority is found among the user appeal, but with the advent of cloud calculation and by using this executed prioritization, the cloud admin may take decisions without difficulty based on the derived maximized values for an efficient assigning of the possible resources. The users with greater priority will be provided the initial chance of functioning, while those with less priority will be considered subsequently. In this method, the cloud administrator can effectively assign the resources to the users with less wastage and proffers optimum profit.

4. Results and discussion

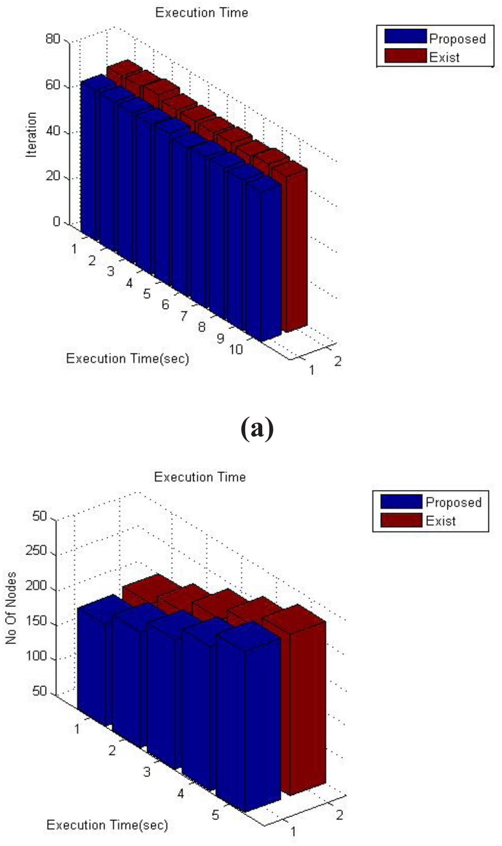
The recommended approach in this study guarantees a powerful network resource management through a reduction of the energy consumption and execution time. Moreover, their computational intricacy and computing capacity of processing elements were also reduced. The recommended methodology was executed on a MATLAB (version 14) platform with the machine configuration as follows: OS: Windows 7, CPU Speed: 3.20 GHz, Processor: Intel Core i7, RAM: 4GB. The performance of the suggested technique was matched with that of other prevailing techniques.

4.1 Performance analysis

In this section, the performance of the suggested method was analyzed using various measures such as the execution time (by varying the nodes and iterations), energy loss (varying the nodes and iterations), node loss (by varying the iterations and reputation) and benchmarking with the prevailing methodologies. The performance of the proposed method is presented in Figures 4 to 8 and are described below.

4.1.1 Execution time

The difference between the time of initializing a task and the time of its completion is the implementation time of the individual task. The estimated execution time of a task is necessary to be determined prior to the decision of whether to offload the task or not, especially for the compute-intensive tasks. The node-based and iteration-based implementation time of the recommended task in this study were matched with the prevailing ABC using the method previously described and the outcome is presented in Figure 4 (a, b). The outcome of the comparison proved that the suggested task implementation time was less than that of the prevailing techniques.



(a)

(b)

Figure 4 (a, b). Comparison analysis of suggested CS with prevailing ABC for Execution time

4.1.2 Reputation

Reputation is the complete quality or peoples' character generally observed or evaluated. In networks, status is the global perception of a node's trustworthiness. The graphical illustration of the reputation of the recommended methodology based on varying the iterations is presented in Figure 5.

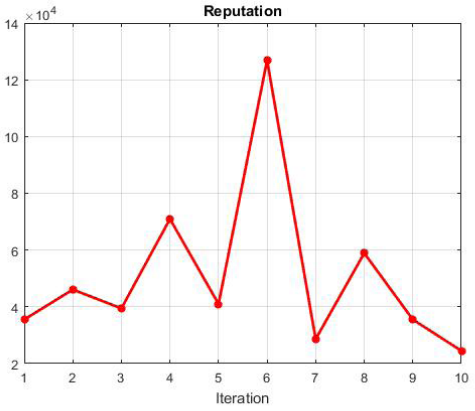


Figure 5. The graphical illustration of suggested reputation by varying iterations

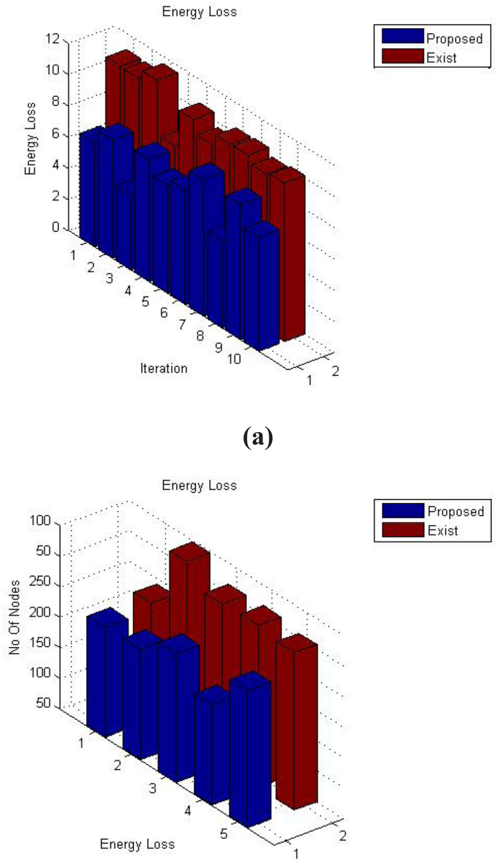
4.1.3 Energy loss

The energy loss E_{loss} within a network can be estimated by multiplying the energy loss factor with the load losses at the peak load and period T, then, added to the no-load losses multiplied by the same period. It can be calculated using Equation 17,

$$E_{loss} = F * T * P_{load\ loss\ peak} + T * P_{no\ load\ loss}, \quad (17)$$

where E_{loss} is the energy loss, F is the loss factor, T is the time period, $P_{load\ loss\ peak}$ is the load losses at peak load, $P_{no\ load\ loss}$ is the no-load losses.

The comparative analysis of the energy loss of the proposed methodology with the existing methods based on varying the number of nodes and iterations is presented in Figure 6. The result proved that the suggested task implementation time was less than those of the prevailing methodologies.



(a)

(b)

Figure 6 (a, b). Comparative analysis of the suggested CS with the prevailing ABC for energy loss

Figure 6 (a, b) showed the comparison graph of the proposed CS energy losses with that of the prevailing ABC optimization techniques. The node-related and iteration-based energy losses were matched with the prevailing method and the results proved that the energy loss in the recommended model was less compared to that of the prevailing techniques.

4.1.4 Delay

The network delay is the time taken for a data to travel from one node or endpoint to another through a network. It is generally determined in multiples or fractions of a second. Delay may slightly vary based on the location of the pair of interacting nodes. The delay in data transmission is proportional to the length of the data (in bits); it is estimated using the formula below.

$$T_{delay} = {}^{N}\!/_{R} \tag{18}$$

D$_T$ where T_{delay} D$_T$ is the transmission delay in seconds, N is the number of bits, and R is the rate of transmission (say in bits per second). The proposed work delay for varying number of iterations is presented in Figure 7.

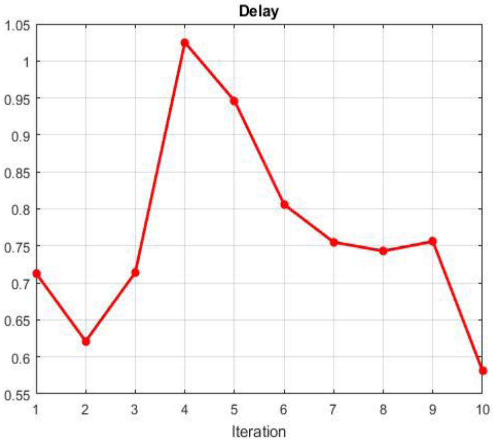


Figure 7. The graphical depiction of the suggested delay by varying the iterations

4.1.5 Node loss

When a node fails, the only sign has been the integration loss to the node processes as observed by other nodes. Thus, nodes are regarded as failed when their membership is lost with a significant component. That is, when the nodes that create the primary feature cannot observe the node anymore, that node has failed. There was a reduced node loss with the recommended technique in this study based on different iterations, as presented in Figure 8.

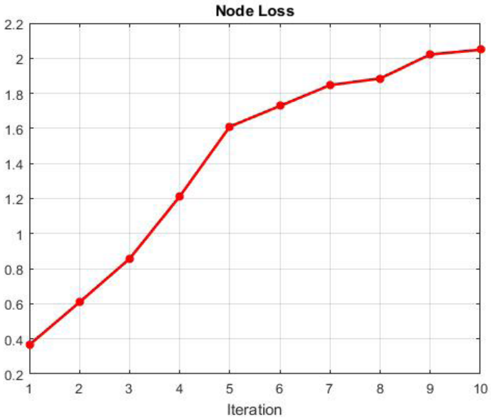


Figure 8. The graphical illustration of node loss by varying the iterations

5. Conclusion

This paper emphasized a powerful resource management scheme accompanied by a minimal energy consumption through the utilization of the enthalpy-related CSO algorithm. We formulated and rectified the resource management issues in this task; for the first task, different effective measures were computed for each task and based on the computed enthalpy values, the CSO algorithm prioritized the resources optimally. Lastly, the suggested enthalpy-related CSO prioritized the tasks based on the maximum outcomes and through which, the resources were planned in an effective manner. The execution of the suggested resource management scheme was implemented on a MATLAB platform. The performance of the recommended management scheme was benchmarked with that of typical Artificial Bee Colony (ABC) algorithm. The energy consumption and calculated time of the proposed scheme were considerably lowered. These findings rendered the proposed scheme as a viable and precious procedure for resource allocation optimization.

REFERENCES

1. Abu Sharkh, M. & Shami, A. (2017). An evergreen cloud: Optimizing energy efficiency in heterogeneous cloud computing architectures, *Vehicular Communications*, *9* (Supplement C), 199-210.

2. Ahmad, A., Paul, A., Khan, M., Jabbar, S., Rathore, M. M. U., Chilamkurti, N. & Min-Allah, N. (2017). Energy Efficient Hierarchical Resource Management for Mobile Cloud Computing, *IEEE Transactions on Sustainable Computing*, *2*(2), 100-112.

3. Benkhelifa, E., Welsh, T., Tawalbeh, L., Jararweh, Y. & Al-Ayyoub, M. (2016). Leveraging software-defined-networking for energy optimisation in mobile-cloud-computing, *Procedia Computer Science*, *94*, 479-484.

4. Bermejo, B., Guerrero, C., Lera, I. & Juiz, C. (2016). Cloud Resource Management to Improve Energy Efficiency Based on Local Nodes Optimizations. In *Procedia Computer Science*, *83*(*Supplement C*), 878-885.

5. Bui, D.-M., Yoon, Y., Huh, E.-N., Jun, S. & Lee, S. (2017). Energy efficiency for cloud computing system based on predictive optimization, *Journal of Parallel and Distributed Computing*, *102*(Supplement C).

6. Cojoacă, E. Ș. D., Popescu, M. A.-M. & Ambăruș, G. C. (2017). Cloud Computing Technology to Assist Government in Decision Making Process, *Studies in Informatics and Control*, *26*(2), 249-258.

7. Duan, H., Chen, C., Min, G. & Wu, Y. (2017). Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems, *Future Generation Computer Systems*, *74*(*Supplement C*), 142150.

8. Gai, K., Qiu, M., Zhao, H., Tao L. & Zong Z. (2016). Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing, *Journal of Network and Computer Applications*, *59*, 46-54.

9. Ghasemi-Falavarjani, S., Nematbakhsh, M. & Ghahfarokhi, B. S. (2015). Context-aware multi-objective resource allocation in mobile cloud, *Computers & Electrical Engineering*, *44*, 218-240.

10. Ghomi, E. J., Rahmani, A. M. & Qader, N. N. (2017). Load-balancing Algorithms in Cloud Computing: A Survey, *Journal of Network and Computer Applications*.

11. Hasan, R. A. K., Mohammed, M. A., Ţăpuş, N. & Hammood, O. A. (2017). A comprehensive study: ACO for Facility Layout Problem. In *16th RoEduNet Conference: Networking in Education and Research*, 1-8.

12. Kaewpuang, R., Niyato, D., Wang, P. & Hossain, E. (2013). A framework for cooperative resource management in mobile cloud computing, *IEEE Journal on Selected Areas in Communications*, *31*(12), 2685-2700.

13. Kumar, N., Zeadally, S. & Misra, S. C. (2016). Mobile cloud networking for efficient energy management in smart grid cyber-physical systems, *IEEE Wireless Communications*, *23*(5), 100-108.

14. Liang, H., Cai, L. X., Huang, D., Shen, X. & Peng, D. (2012). An SMDP-based service model for interdomain resource allocation in mobile cloud networks, *IEEE transactions on vehicular technology*, *61*(5), 2222-2232.

15. Liu, T., Chen, F., Ma, Y. & Xie, Y. (2016). An energy-efficient task scheduling for mobile devices based on cloud assistant, *Future Generation Computer Systems*, *61*, 1-12.

16. Liu, Y. & Lee, M. J. (2015). An adaptive resource allocation algorithm for partitioned services in mobile cloud computing. Paper presented at the *2015 IEEE Symposium Service-Oriented System Engineering (SOSE)*.

17. Liu, Y. & Lee, M. J. (2015). Security-aware resource allocation for mobile cloud computing systems. Paper presented at the *2015 24th International Conference on Computer Communication and Networks (ICCCN)*.

18. Luong, N. C., Wang, P., Niyato, D., Wen, Y. & Han, Z. (2017). Resource management in cloud networking using economic analysis and pricing models: a survey, *IEEE Communications Surveys & Tutorials*, *19*(2), pp. 954-1001.

19. Manvi, S. S. & Shyam, G. K. (2014). Resource management for Infrastructure as a Service (IaaS) in cloud computing: A

survey, *Journal of Network and Computer Applications*, *41*, 424-440.

20. Merezeanu, D., Vasilescu, G. & Dobrescu, R. (2016). Context-aware Control Platform for Sensor Network Integration in IoT and Cloud, *Studies in Informatics and Control*, *25*(4), 489-498.

21. Mohammed, M. A. & Hasan, R. A. K. (2017). Particle Swarm Optimization for facility layout problems FLP – A comprehensive study. In *IEEE 13th International Conference on Intelligent Computer Communication and Processing*, *(ICCP) 2017*, 1-7.

22. Nawrocki, P. & Reszelewski, W. (2017). Resource usage optimization in Mobile Cloud Computing, *Computer Communications*, *99*, 1-12.

23. Rahimi, M. R., Ren, J., Liu, C. H., Vasilakos, A. V. & Venkatasubramanian, N. (2014). Mobile cloud computing: A survey, state of art and future directions, *Mobile Networks and Applications*, *19*(2), 133-143.

24. Ren, S. & Van der Schaar, M. (2014). Dynamic scheduling and pricing in wireless cloud computing, *IEEE Transactions on Mobile Computing*, *13*(10), 2283-2292.

25. Tang, L. & Chen, H. (2017). Joint pricing and capacity planning in the iaas cloud market, *IEEE Transactions on Cloud Computing*, *5*(1), 57-70.

26. Tawalbeh, L. A., Jararweh, Y. & Dosari, F. (2015). Large-scale cloudlets deployment for efficient mobile cloud computing, *Journal of Networks*, *10*(1), 70-77.

27. Vakilinia, S., Ali, M. M. & Qiu, D. (2015). Modeling of the resource allocation in cloud computing centers, *Computer Networks*, *91*, 453-470.

28. Varghese, B. & Buyya, R. (2017). Next generation cloud computing: New trends and research directions, *Future Generation Computer Systems 1*.

29. Yang, J., Lu, Z., Wang, N., Wu, J. & Hung, P. C. (2017). Multi-policy-aware MapReduce resource allocation and scheduling for smart computing cluster, *Journal of Systems Architecture*, *80*, 17-29.