Energy Consumption Reduction Strategy and a Load Balancing Mechanism for Cloud Computing in IoT Environment

Tai Zhang, Huigang Li
Hebei Software Institute
Hebei Baoding 071000, China

***Abstract*—Modern networks are built to be linked, agile, programmable, and load-efficient in order to overcome the drawbacks of an unbalanced network, such as network congestion, elevated transmission costs, low reliability, and other problems. The many technological devices in our environment have a considerable potential to make the connected world concept a reality. The Internet of Things (IoT) is a research community initiative to bring this idea to life. Cloud computing is crucial to making it happen. The load balancing and scheduling significantly increase the possibility of using resources and provide the grounds for reliability. Even if the intended node is under low or high loading, the load balancing techniques can increase its efficiency. This paper presents a scheduling technique for optimal resource allocation with enhanced particle swarm optimization and virtual machine live migration technique. The proposed technique prevents excessive or low server overloads through optimal allocation and scheduling tasks to physical servers. The proposed strategy was implemented in the cloudsim simulator environment and compared and showed that the proposed method is more effective and is well suited to decreasing execution time and energy consumption. This solution provides grounds to reduce energy consumption in the cloud environment while decreasing execution time. The simulation results showed that the amount of energy consumption compared to particle crowding has decreased by 10% and compared to PSO (Particle Swarm Optimization) scheduling by more than 8%. Also, the execution time has been reduced by 18% compared to particle swarm scheduling and by 8% compared to PSO.**

Keywords—Internet of things; load balancing; cloud computing; virtual machine migration

I. Introduction

Cloud computing is a computing model based on large computer networks such as the Internet, which provides a new model for the supply, consumption, and delivery of information technology services (including hardware, software, information, and other shared computing resources) using the Internet. Naturally, every change and new concept in the world of technology has its own advantages, problems, and complications [1]. Using cloud computing is not an exception to this rule. Among the advantages of cloud computing, we can mention the lack of time and place restrictions, simple sharing of resources, as well as the reduction of capital and operational costs (the most important advantage), because in fact, cloud computing dynamically provides scalable resources as a service on the Internet and has also put many challenges in front of experts in this field, among which we can mention things like: resource allocation and load balancing, security, reliability, ownership, data backup and data portability [2, 3, 34]. Meanwhile, resource allocation and load balancing in cloud computing are of great importance. This issue has been discussed in various fields such as operating system, cluster computing and data center management. A resource allocation system in cloud computing can be considered as any mechanism whose purpose is to ensure that the requirements of applications are met [4, 36]. In addition, the resource allocation mechanism must examine the current status of each resource in the cloud environment in order to provide algorithms for better allocation of physical resources or virtual resources and thus reduce operational costs in the cloud environment. It is clear that due to the scale and complexity of these systems, the centralized assignment of tasks to a specific server without considering specific solutions is actually impossible, and also due to the increasing load and volume of requests in advanced data centers and the urgent need to achieve quality. For optimal service, the need for solutions to increase the efficiency of existing servers in the data center is felt. One of the ways to achieve optimal productivity is to use scheduling and load balancing solutions [5, 33]. This technology, with its high potential of cloud computing for storing and processing data remotely, has provided a new computing model. Recently, instead of using domestic resources, many large companies have outsourced them to cloud computing [6]. So users can access their data anywhere in the world, and they do not need high-performance hardware and storage systems because all computing and storage operations are performed by cloud service providers and well-equipped and advanced servers. Meanwhile, the scheduling and resource allocation problem in cloud computing is important because it directly affects the amount of energy consumption and reduction of latency in service provision [7]. A scheduling system in cloud computing can be any mechanism to ensure the provision of application requirements. In addition, the scheduling mechanism should examine the current status of each source in the cloud environment to provide algorithms for better allocation of physical or virtual resources and thus reduce operational costs in the cloud environment. That is because, during the processing, a number of servers might have a high traffic load. During the load distribution among servers with less load, the idle servers can be turned off to reduce energy consumption [8, 35, 26]. In this study, a PSO algorithm is suggested for scheduling and ideal load balancing in the cloud infrastructure in order to reduce energy usage in cloud computing environments' data centers. The proposed technique prevents server overloads or low load through the optimal assignment of tasks to physical servers. This research is also important from another aspect that with the increase of users and their different requests, the following situations may occur:

- The virtual machine may be performing an operation and not accept another request.
- The request should be made on a new car.
- The request should be applied to a machine that is busy and does not have enough capacity to receive new ones.
- Migration may take place.

Therefore, a solution must be provided to manage these challenges. When the bandwidth of a virtual machine is full, the central cluster sends requests to another machine. At this stage, an algorithm is needed to balance the load and choose the right processing server. Based on this, a solution based on the improved particle community optimization algorithm is presented for efficient scheduling and as a result optimal load balancing in the cloud infrastructure, so that the task execution time can be reduced by broadcasting the requests and in As a result, it helped to manage energy consumption for users. In this research, as an innovation, for the purpose of scheduling, a combined method of particle swarm optimization and virtual machine live migration technique has been used to balance the load using an optimal schedule. The use of combined methods for load balancing has not been much considered in other researches. Also, the goals and contributions of the authors in this research are stated as follows:

- Creating load balance and reducing response time to users' requests.
- Balancing load in cloud network using particle swarm algorithm and task migration.

In the continuation of our paper, it is configured as follows. The next section, which discusses prior works in the area of cloud computing in the IoT context, presents a list of related works. In this section's conclusion, the current works' characteristics are given in table format for different criteria. Section III describes the proposed approach in three parts: problem-solving formulation, H algorithm based on scheduling, and virtual machine migration. The performance evaluation for energy consumption parameters and execution time is given in Section IV. The analysis of the comparison between the suggested approach and the current works is presented as a table in the fifth part. In the last part, the conclusion of the research is given.

II. Previous Works

This section looks at concerns like access control and load balancing for cloud computing networks. Also, the discussion related to the Internet of Things environment is covered. A method based on efficient workload distribution and resource management is suggested in reference [9] employing a cloud computing framework. In this approach, clustering learning techniques are employed to decrease network edge energy consumption as well as processing and communication delays. A similar strategy is presented in reference [10] to build a true edge cloud ecosystem. In the Internet of Things, a capillary computing architecture for orchestrating microservices from edge devices to cloud computing providers is suggested in reference [11]. An Edge-Fog-Cloud environment is described as a distributed cloud for IoT computing in reference [12]. The authors also review a cloud computing offloading strategy for simultaneous localization and mapping of indoor mobile robots in reference [13]. A trust management technique is described in reference [14] to enhance a distinct perspective on the cloud environment and to enable a blockchain-based cloud computing architecture. A safe offloading technique based on machine learning is also suggested in reference [15] for cloud computing, which expands the potential of IoT for applications related to smart cities. A semantic model-based strategy for IoT data description and discovery for IoT-Cloud architecture is presented in reference [16]. They suggested a method for optimizing energy consumption in a set of heterogeneous computing groups to serve various web applications in one of the earliest studies in scheduling and power management [17]. The suggested method decides whether to turn on or off the nodes to reduce total energy consumption while controlling the resources sporadically. In order to reduce power consumption, the request scheduling issue for multi-layer web applications was researched in [18]. It was suggested in [19] to use Power-Aware Tasks Scheduling (EATS) to divide and schedule vast amounts of data in the cloud. This model's primary objective was to improve application efficiency and lower energy usage in subsurface resources. According to [20], the grouped tasks scheduling (GTS) algorithm is used to plan out the tasks in a network of cloud computing services while considering customer needs for service quality. The suggested algorithm creates five groups to split tasks belonging to each group and share characteristics such as user type, task type, job size, and work delay. In order to run task-based efficient programs on distributed operating systems and save energy, a real-time dynamic scheduling system was developed [21]. There is currently no optimal multiprocessor solution for the NP-hard job scheduling problem. Therefore, a polynomial algorithm is suggested in [22] that combines exploratory principles and resource allocation strategies to locate suitable solutions quickly. Table I provides a comparison of the aforementioned techniques. The comparison is based on the purpose, the preferred method, and the result of the work.

TABLE I. REVIEW OF PREVIOUS WORKS IN THE FIELD OF LOAD BALANCING USING CLOUD COMPUTING

Ref.	**Objective**	**Method Referred**	**Achievement**	**Inference**
[1]	Utilizing cloud computing for IoT resource allocation and workload distribution	Learning classifier	40% reduction in processing delays	The delay in the transmission of packets is reduced. Reducing energy consumption
[2]	Resource management in cloud computing	Pseudo code-dynamic testing	Increasing throughput and reducing latency	• This solution is affordable, scalable and reliable.
[3]	Developing a new architecture for smart applications that support different IoT workloads in the cloud computing environment.	orchestration based on containers	Many times faster for response time	• A Fog or Cloud resource was successfully offloaded from an Edge node. • Capable of coping with extremely dynamic IoT situations.
[4]	Distributed tasks processing in Cloud computing environment	The lowest processing cost is used in the first method to assign tasks.	Extend time without compromising associated costs	• Display processing and network costs. • Evaluation of Edge, Fog, and cloud computing options in light of device connection congestion.
[5]	Secure offloading for Cloud Computing of things	Machine learning methods, fuzzy neural model	Reduce latency	Cloud selection is based on reinforcement learning, and cloud node availability is estimated using available processing power and remaining node energy.
[6]	Internet of Things data management methods in IoT-Fog-Cloud	Better characteristics of IoT data flow in semantic model	IoT data stream characterization to support semantic data retrieval	• Attention to data storage issues. • The creation of protocols for data discovery for Internet of Things hardware.
[7]	For large-scale Internet of Things systems, enabling efficient access control procedures.	Token management methods with identity-based capabilities	An approach to access management for Internet of Things systems that is scalable	• Achievements of capabilities such as load balancing, decentralized access and lightweight approach
[8]	presenting a framework for predictive analytics using the IoT for mobile devices	utilizing machine learning to analyze data	Data privacy, low cost of data transit to data centers, and quick feedback	• Big data management for IoMT devices

III. PROPOSED APPROACH

This section provides the particle swarm optimization algorithm, and the virtual machine lives migration technique to create optimal load balancing in the cloud infrastructure. Using this solution, the execution time of the tasks decreases, and the energy consumption is also reduced. Accordingly, the formulation of the solution is presented below.

A. Formulation of the Solution

Since mapping the workflow of a program to distributed resources can have several goals, the present study is focused on two goals: energy consumption and traffic consumption. The two targets are formulated as follows.

Studies in [23] have demonstrated that a linear relationship between energy consumption and CPU sufficiency can correctly describe the energy consumption of servers. As a result, the following model describes how much energy a physical machine uses in a cloud environment:

$$\mathrm{P}_j = \begin{cases} \left(\mathrm{P}_j{}^{busy} - P_j{}^{idle}\right) \times U_j{}^{p} + P_j{}^{idle}, & \mathrm{U}_j{}^{c} > 0 \\ 0 & , \text{ otherwise} \end{cases} \tag{1}$$

Where:

P_j: Energy consumption of the physical machine j

$\mathrm{P}_j{}^{idle}$: The average energy consumption of the physical machine j when it is idle

P_j^{busy}: The average energy consumption of the physical machine j when it is busy

$U_j{}^{p}$: The normalized amount of processor consumed by the physical machine j

$\mathrm{U}_j{}^{c} > 0$: The physical machine j is on

The main idea behind this kind of modeling is to convert the less-busy physical machines into idle and then turn them off. In this relation,$\mathrm{P}_j{}^{busy}$ and $P_j{}^{idle}$ are constant values which are 162 and 215 watts for Dell physical machines.

The following equation is used to calculate the network bandwidth consumption and traffic generated by each physical machine:

$$D_j = \sum_{\forall m \epsilon V_j} \lambda(j,m) \sum_{i=1}^{\rho(j,m)} C_i \tag{2}$$

Where:

D_j: The communication between the physical machine j and other physical machines

$\lambda(j\,\text{و}m)$: Traffic load between the physical machine j and other physical machines

V_j: The set of physical machines communicating with the physical machine j

C_i: The weight of communication link between two physical machines at level i

$\rho\left(j\,\text{و}m\right)$: The level of communication between the physical machines m and the physical machine j

Moreover, the approach presented in [24] is used to model resource usage. The model is predicated on the idea that the devices' power consumption and processor productivity truly follow a linear relationship. To put it another way, the knowledge about a task's processing time and CPU efficiency is sufficient to determine its power usage. The definition of efficiency for a resource like ri at any given time is as follows:

$$U_i = \sum_{j=1}^{n} u_{i,j}$$

Where n is the number of currently active tasks and uij is the number of resources task tj uses. As a result, the following formula is used to determine the energy consumption (Ej) of the rj resource:

$$E_j = (P_{max} - P_{min}) \times U_i + P_{min} \tag{3}$$

In this case, Pmax is the maximum power consumption (i.e., at 100% efficiency), and Pmin is the minimum energy consumption per minute when the server is active (or at 1% efficiency).

B. Load Balancing Processes

Based on the above strategies, the main steps for load balancing are as follows:

- Calculating the transfer probability for all its neighbors and selecting the largest one as the next destination.
- Moving to a new node and checking if it is a candidate node; if the answer is yes, a migration should be created and initialized. For advanced migration, you have to go back to the first step.
- Retrograde migration returns to the starting point of the corresponding leading ant and in the same direction as the leading migration. During the route, the information related to the pheromone of each node that the backward migration passed through was updated and if it reached the starting point, it deleted the backward migration.
- Calculating the total resources of the candidate nodes and if the nodes need to perform load balancing operations, these steps will be stopped.
- Finally, in the last step, the load balancing operation should be performed using the live migration of the virtual machine.
- These steps for max-min rules also; it is similar except in the step of calculating the transfer probability.

C. Particle Swarm Optimization based Scheduling

Now, based on the relations presented in the previous section using the PSO algorithm for scheduling is discussed. Other evolutionary algorithms and the PSO algorithm are comparable. The population in PSO is the same as the number of particles in the problem space. Initialization of the particles is random. Every particle will have a compatibility value, which will be evaluated by the compatibility function, which ought to be optimized for every generation. This algorithm attempts to identify the best solution by updating generations after first creating a collection of particles at random [25]. In actuality, each particle chooses the best location (pbest) and position from the gbest particles. Each particle's pbest represents its best outcome to date, while its gbest represents its best compatibility with the entire population. Each generation will update the particle positions and velocities using relations (4) and (5), respectively:

$$v[\,] = v[\,] + \tag{4}$$

c1 * rand() * (pbest[ ] - position [ ]) +

c2 * rand() * (gbest[ ] - position[ ])

$$position[\,] = position[\,] + v[\,] \tag{5}$$

The first part of equation (4) is the particle's current velocity. In contrast, the second and third portions are the particle's change in velocity and its rotation toward the best individual experience and collective experience, respectively. The initialization of the particle position and velocity in the PSO algorithm is random. The given tasks in this study are the particles, and the size (dimension) of the particles represents the total number of tasks in the workflow. The processing resource indices are the values assigned to each particle dimension. Each particle, then, represents the mapping of a resource to a job. Each particle is assessed using the compatibility function provided in relations (1) and (2). Equation (3) is used to calculate the velocity of particles, while Equation (3) is used to update their location (4). This assessment continues until it reaches a certain number of repetitions.

The PSO-based scheduling algorithm is presented in Fig. 1. For scheduling by PSO algorithm, the initial step is calculating the traffic cost between resources using equation (2) based on the current network load. The algorithm then distributes completed tasks to resources by PSO's mapping. Then, according to the number of accomplished tasks, the list is ready to be updated, including tasks that their father completes. Following that, the energy wastage cost is calculated. These steps continue until all tasks are scheduled in the workflow.

```
Calculate the cost of traffic between resources using equation (2)
Compute PSO ({Task})
for all “ready” tasks {Task} ∈ T do
Assign tasks to resources sby PSO
Dispatch all the mapped tasks
Update task list
Calculate energy wastage cost using equation (1)
Compute PSO ({Task})
Perform VM Migration for load balancing.
```

Fig. 1. PSO Scheduling Algorithm.

D. Virtual Machine Migration

During the execution process of scheduling tasks on processing servers performed with the help of particle swarm optimization, some servers may fail to execute tasks for reasons such as delays in previous processing tasks. They might have additional traffic load due to the delay in processing tasks, or hardware and software errors may cause additional traffic and overload while some servers are idle. In this case, the processing servers are reviewed periodically, and then the following three conditions are used to determine the status of the processing servers:

- If the server processor achieves its maximum productivity, then the server falls into the category of overloaded machines (Poverload).
- Suppose the server processor productivity is lower than the specified average productivity. In that case, it falls into the category of underload servers, which is usually around 10%. (Punder)
- Other servers that do not meet any of the above requirements are included in the normal server (Pnormal).

To make load balancing, the overload servers should migrate to under-load servers, so loads of servers in overload mode are migrated through the virtual machine migration technique onto another server in Punder or normal mode. Other servers with productivity below 10 percent will be candidates for tasks migration, and therefore they are turned off to reduce power consumption.

IV. Analysis, Evaluation and Performance

A model that could simulate and evaluate the cloud infrastructure equipment is required to simulate and evaluate the solution. CloudSim is an extendable simulation tool that enables modeling and simulating cloud computing systems and preparing applications. In reality, this program makes it possible to do integrated modeling, simulation, and assessment of cloud infrastructure and associated services [26]. In order to simulate data centers, 800 hosts have also been defined. A class called a server that expands the Power Model SpecPower class, a subclass of the Power Model class in CloudSim, must be developed in order to establish any virtual server in the software. The CoMon project, a monitoring infrastructure for PlanetLab, has tested real traced data, which has been used to generate the workload. The Round-Robin and ant colony optimization methods are compared to the strategy. The power consumption, execution time, and SLA criteria are taken into account to evaluate the suggested algorithm's effectiveness compared to other methods. Finally, the evaluation results are shown in Fig. 2 to 6.

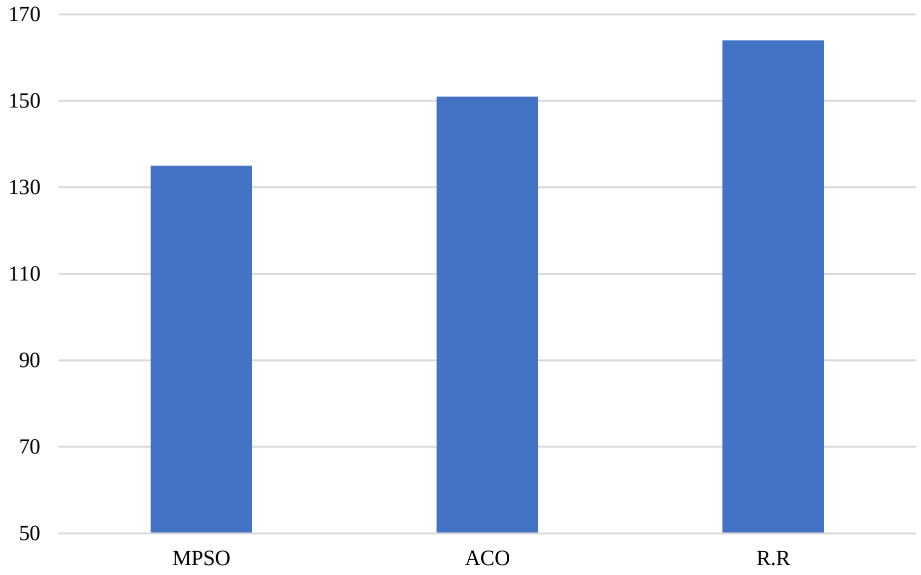

Fig. 2. Energy Consumption (kw/h).

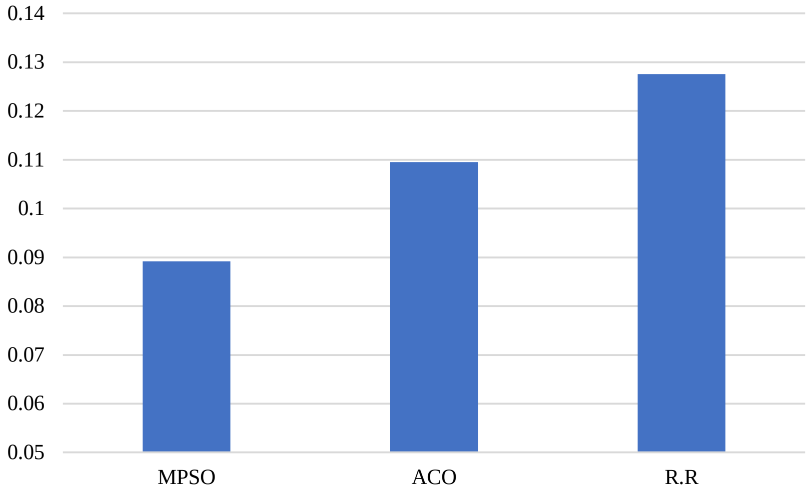

Fig. 3. Execution Time (Second).

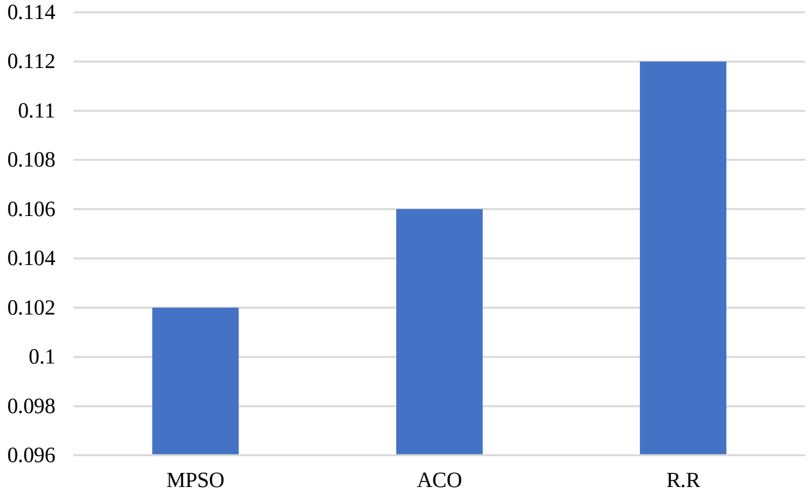

Fig. 4. Service Level Agreement (%SLA).

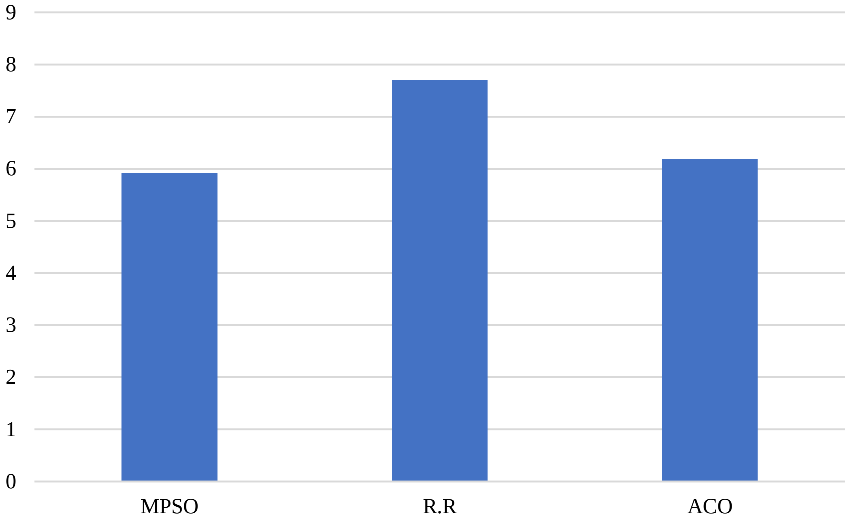

Fig. 5. Service Level Agreement Degradation due to Migration (%SLA).

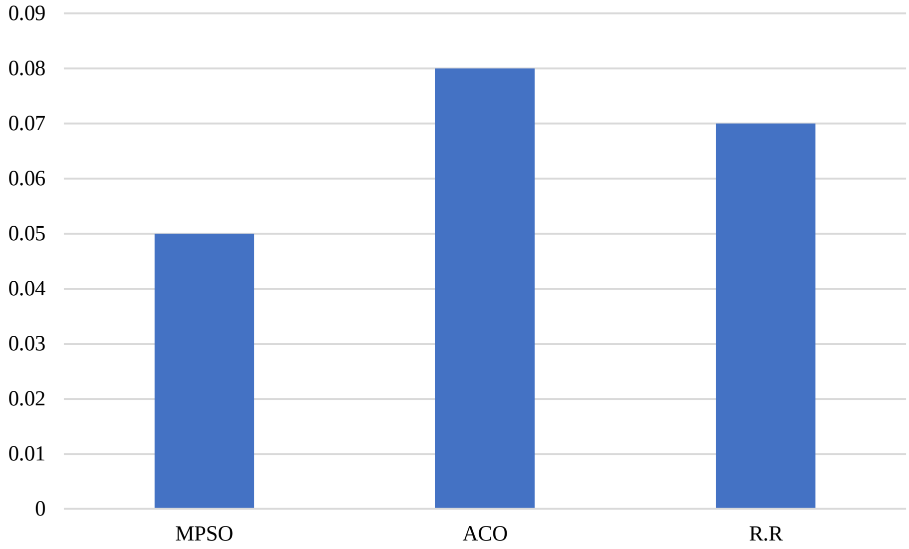

Fig. 6. Execution Time - VM Reallocation Mean.

According to Fig. 2, the energy consumption in the proposed solution is greatly reduced compared to the other two solutions due to appropriate scheduling and balanced load distribution between processing servers and turning the idle servers off. Accordingly, the scheduling is reduced by 10 and 17% compared to the Ant Colony optimization and Round Robin algorithms. Given the number of resources and energy consumption in each schedule, this reduced energy consumption was predictable.

In addition to reducing energy consumption, reducing demand execution time is also very important in cloud infrastructure. In fact, it is another goal of this research. Accordingly, the execution time in the solution has been reduced to an appropriate level, as shown in Fig. 3. The proposed solution has reduced the execution time by 18 and 28% more than the R.R and ant colony optimization, respectively. This is due to load balancing in the cloud infrastructure by utilizing PSO-based scheduling and the task execution predictor. Moreover, the use of the virtual machine live migration technique allows transferring the processing load to free or idle servers, and the execution time is reduced significantly.

In Fig. 4, the service level agreement violations have been shown. The Service Level Agreement (SLA) is the basis for determining the expected level of service. The service quality parameters in the service level agreement specify the extent to which the provided quality is appropriate. Customers need this contract to ensure the quality level of their services. In fact, the primary purpose of this agreement is to define an official basis for the terms of provided service, such as efficiency or availability. According to the figure, using the load balancing produced by the optimal scheduling, the proposed strategy has provided the servers with higher reliability than the other two strategies, creating better availability in the cloud infrastructure. In other words, the violation percentage of this solution is lower than the service level agreement compared to the other two solutions.

According to Fig. 5, the proposed solution has the least SLA degradation due to migration because all migrations are optimized and adapted to the needs of the infrastructure resulting in the lowest reduction in quality of service. In fact, the service quality parameters in the service level contract determine how appropriate the quality of the provided service is. Customers ensure the quality of the provided service through this contract. As seen in the figure, thanks to the load balancing created by optimizing the timing, the proposed solution has made the servers more reliable than the other two solutions. It thus has higher availability in the cloud infrastructure compared to other solutions. In other words, the percentage of violations of this solution is less than the service level agreement compared to the other two solutions.

Since virtual machine migration can affect runtime, Fig. 6 shows the VM reallocation mean time required for virtual machine reallocation. As it can be seen, the proposed solution in this parameter is also more efficient than other methods. In other words, since this solution examines the status of server resources and allocates them accordingly, it creates the least delay in executing user requests.

V. Comparative Analysis

Table II describes the comparative analysis of the suggested method with related research. The comparison is based on features of already published literature, including load balancing, resource management, and energy usage reduction. Almost all publications emphasize the significance of load balancing, except [23, 27] and certain works that allude to resource management, including [19, 20, 22, 23, 25]. Almost all of these studies have not covered the idea of lowering energy usage and scheduling. The references [20, 21, 24, 25] have used edge computing methodologies. They also employ cloud computing in many aspects of their work.

TABLE II. COMPARING THE SUGGESTED APPROACH WITH THE CURRENT APPROACHES

Ref.	Resource management	Load balancing	energy consumption reduction	Scheduling based	Cloud based
[7]	✓	✓	×	×	✓
[8]	✓	✓	×	×	✓
[9]	✓	×	×	✓	✓
[27]	×	✓	×	×	✓
[28]	✓	×	×	✓	✓
[29]	✓	×	×	×	✓
[30]	×	×	×	×	✓
[31]	✓	✓	×	×	✓
[32]	✓	×	×	×	✓
Proposed approach	✓	✓	✓	✓	✓

Also, this paper presents a scheduling technique for optimal resource allocation with particle swarm optimization and virtual machine live migration technique. The proposed technique avoids server overload or underload through optimal allocation and scheduling of tasks to physical servers. The proposed strategy was implemented and compared in the cloudsim simulator environment and showed that the proposed method is more effective and suitable for reducing execution time and energy consumption. This solution provides the basis for reducing energy consumption in the cloud environment and at the same time reducing execution time. Compared to the previous test, due to the increase of ants, it is clearly visible that the virtual machine migration is increasing compared to the previous state which has caused the execution time to decrease. In fact, by applying suitable and optimal migrations, additional load has been transferred from high traffic servers to low traffic servers. As can be seen from Table III, although the amount of energy consumption has increased with the increase in the number of ants, but at the same time, the amount of energy consumption has decreased by approximately 12% compared to the MPSO (Multi_Objective Particle Swarm Optimization) method and 9% compared to Round Robin. Also, by placing Optimum methods and reduction of additional overhead, execution time has also been optimized compared to all three other methods. As compared to the R.R. method, it has reached 27%.

TABLE III. THE DEGREE OF OPTIMALITY OF THE PROPOSED SOLUTION COMPARED TO OTHER SOLUTIONS (100 ANTS)

	PSO	R.R	Random
energy consumption	12%	11%	5%
Virtual machine migration	12%	13%	17%
execution time	5%	29%	16%

VI. CONCLUSIONS

Due to the scale and complexity of the cloud infrastructure, the centralized assignment of tasks to a specific server without considering any specific solutions is impossible. Also, considering the increasing workload and volume of requests in advanced data centers and the urgent need to achieve optimal service quality, solutions should be developed to increase the productivity of existing servers in the data center. One method to achieve optimal productivity is the use of scheduling technics. Load balancing is needed to manage the service providers' resources properly. This study uses a solution based on a swarm particle optimization algorithm and the virtual machine lives migration technique for load balancing. The proposed technique prevents server overloads or low load through the optimal assignment of tasks to physical servers. Finally, it was shown with the help of ClodSim software and in the simulation environment that this method significantly increases the efficiency. For this purpose, the proposed solution was evaluated in two cases with the number of ants 50 and 100. In the first test (number of 50 ants), the solution was compared with the three algorithms ACO, Round Robin and MPSO and it was shown that the proposed solution was able to execute the desired requests in less time than the two algorithms ACO and Round Robin. Efficiency can be provided with the help of the correct migrations that the solution has provided, and the machines that have had a heavy load have been transferred from them to another machine that has a lower load, and on the other hand, the machines that have no load have also been transferred, found correctly and by turning them off, the amount of energy consumption is also reduced; so that compared to the ACO algorithm, the energy consumption has decreased by almost 10% and compared to the round robin algorithm by 13%, but compared to the MPSO, the optimization in terms of energy consumption has not been achieved. In the second experiment, the number of ants increased to 100 ants and in comparison, it was shown that although the reduction of energy consumption is less than the first case, but with optimal virtual machine migrations, the execution time for all three algorithms is between 12 to 27% decreased. This indicates that the solution is optimal compared to other methods.

In the future, the continuation of this research can be extended to the execution time based on demand resource allocation, because the proposed method does not meet the resource requirements during the execution of requests.

References

[1] Kaur, M., & Aron, R. (2020). Energy-aware load balancing in fog cloud computing. *Materials Today: Proceedings*.

[2] Abdulhammed, O. Y. (2022). Load balancing of IoT tasks in the cloud computing by using sparrow search algorithm. *The Journal of Supercomputing*, *78*(3), 3266-3287.

[3] Maswood, M. M. S., Rahman, M. R., Alharbi, A. G., & Medhi, D. (2020). A novel strategy to achieve bandwidth cost reduction and load balancing in a cooperative three-layer fog-cloud computing environment. *IEEE Access*, *8*, 113737-113750.

[4] Mozaffari, H., & Houmansadr, A. (2020, January). Heterogeneous private information retrieval. In Network and Distributed Systems Security (NDSS) Symposium 2020.

[5] Gowri, V., & Baranidharan, B. (2023). Dynamic Energy Efficient Load Balancing Approach in Fog Computing Environment. In *Intelligent Communication Technologies and Virtual Mobile Networks* (pp. 145-160). Springer, Singapore.

[6] Abdulhammed, O. Y. (2022). Load balancing of IoT tasks in the cloud computing by using sparrow search algorithm. *The Journal of Supercomputing*, *78*(3), 3266-3287.

[7] Javadpour, A., Sangaiah, A. K., Pinto, P., Ja'fari, F., Zhang, W., Abadi, A. M. H., & Ahmadi, H. (2022). An Energy-optimized Embedded load balancing using DVFS computing in Cloud Data centers. *Computer Communications*.

[8] Chapnevis, A., Güvenç, I., & Bulut, E. (2020, November). Traffic Shifting based Resource Optimization in Aggregated IoT Communication. In 2020 IEEE 45th Conference on Local Computer Networks (LCN) (pp. 233-243). IEEE.

[9] Lakra, Atul Vikas, and Dharmendra Kumar Yadav. "Multi-objective tasks scheduling algorithm for cloud computing throughput optimization." Procedia Computer Science 48 (2015): 107-113.

[10] Vahidi Farashah, M., Etebarian, A., Azmi, R., & Ebrahimzadeh Dastjerdi, R. (2021). A hybrid recommender system based-on link prediction for movie baskets analysis. *Journal of Big Data*, *8*(1), 1-24.

[11] Pourghebleh, B., & Hayyolalam, V. (2020). A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things. Cluster Computing, 23(2), 641-661.

[12] Wei, R. (2022). Load Balancing Optimization of In-Memory Database for Massive Information Processing of Internet of Things (IoTs). Mathematical Problems in Engineering, 2022.

[13] Charandabi, S. E., & Kamyar, K. (2022). Survey of Cryptocurrency Volatility Prediction Literature Using Artificial Neural Networks. Business and Economic Research, 12(1).

[14] Ficco M, Esposito C, Xiang Y, Palmieri F. Pseudo-dynamic testing of realistic edge-fog cloud ecosystems. IEEE Commun Mag. 2017;55(11):98-104.

[15] Taherizadeh S, Stankovski V, Grobelnik M. A capillary computing architecture for dynamic internet of things: orchestration of microservices from edge devices to fog and cloud providers. Sensors. 2018;18(9):29-38.

[16] Mohan N, Kangasharju J. Edge-Fog cloud: a distributed cloud for internet of things computations. Proceedings of the 2016 Cloudification of the Internet of Things (CIoT). Paris, France: IEEE; 2016:1-6.

[17] Sarker VK, Queralta JP, Gia TN, Tenhunen H, Westerlund T. Offloading slam for indoor mobile robots with edge-fog-cloud computing. Paper presented at: Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT); May 2019:1-6; Dhaka, Bangladesh: IEEE.

[18] Kochovski P, Gec S, Stankovski V, Bajec M, Drobintsev PD. Trust management in a blockchain based fog computing platform with trustless smart oracles. Futur Gener Comput Syst. 2019;101:747-759.

[19] Madani, M., Lin, K., & Tarakanova, A. (2021). DSResSol: A sequence-based solubility predictor created with Dilated Squeeze Excitation Residual Networks. International Journal of Molecular Sciences, 22(24), 13555.

[20] Zeng W, Zhang S, Yen IL, Bastani F. Semantic IoT data description and discovery in the IoT-edge-fog-cloud infrastructure. Paper presented at: Proceedings of the 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE); April 2019:106-10609; San Francisco, CA, USA: IEEE.

[21] Bouflous, Z., Ouzzif, M., & Bouragba, K. (2023). Analysis of Load Balancing Algorithms Used in the Cloud Computing Environment: Advantages and Limitations. In *Proceedings of the Future Technologies Conference* (pp. 206-226). Springer, Cham.

[22] Trik, M., Molk, A. M. N. G., Ghasemi, F., & Pouryeganeh, P. (2022). A Hybrid Selection Strategy Based on Traffic Analysis for Improving Performance in Networks on Chip. Journal of Sensors, 2022.

[23] Ismail Leila, Fardoun Abbas," Eats: Energy-aware tasks scheduling in cloud computing systems" Procedia Computer Science, V 83, P 870-877, 2016, Elsevier.

[24] S. Pourjabar and G. S. Choi, "A high-throughput multimode low-density parity-check decoder for 5G New Radio," International Journal of Circuit Theory and Applications, 2021.

[25] Sun, J., Zhang, Y., & Trik, M. (2022). PBPHS: A Profile-Based Predictive Handover Strategy for 5G Networks. *Cybernetics and Systems*, 1-22.

[26] Ali Hend Gamal El Din Hassan,Saroit Imane Aly,Kotb Amira Mohamed, " Grouped tasks scheduling algorithm based on QoS in cloud computing network " Egyptian Informatics Journal, V 18, P 11-19, 2017, Elsevier.

[27] Mozaffari, H., Houmansadr, A., & Venkataramani, A. (2019, December). Blocking-Resilient Communications in Information-Centric Networks using Router Redirection. In 2019 IEEE Globecom Workshops (GC Wkshps) (pp. 1-6). IEEE.

[28] Agrawal, N. (2021). Dynamic load balancing assisted optimized access control mechanism for edge-fog-cloud network in Internet of Things environment. *Concurrency and Computation: Practice and Experience*, *33*(21), e6440.

[29] Trick, M., Boukani, B., Emtiyaz, S., Azar, S. R., & Darvandi, F. M. (2014, June). An Overview of through-silicon via–based three dimensional integrated circuits (3D IC) to placement to optimize timing. In International Science Congress Association, Research Journal of Recent Sciences, Manuscript No: ISCARJRS-2013-303.

[30] Pourbemany, J., Mirjalily, G., Abouei, J., & Raouf, A. H. F. (2018, May). Load Balanced Ad-Hoc On-Demand Routing Basedon Weighted Mean Queue Length Metric. In Electrical Engineering (ICEE), Iranian Conference on (pp. 470-475). IEEE.

[31] Rafiee, P., & Mirjalily, G. (2020). Distributed Network Coding-Aware Routing Protocol Incorporating Fuzzy-Logic-BasedForwarders in Wireless Ad hoc Networks. Journal of Network and Systems Management, 28(4), 1279-1315.

[32] Baburao, D., Pavankumar, T., & Prabhu, C. S. R. (2022). A novel application framework for resource optimization, service migration, and load balancing in fog computing environment. Applied Nanoscience, 1-14.

[33] Trik, M., Pour Mozaffari, S., & Bidgoli, A. M. (2021). Providing an adaptive routing along with a hybrid selection strategy to increase efficiency in NoC-based neuromorphic systems. Computational Intelligence and Neuroscience, 2021.

[34] Yamini, R. "Energy aware green task assignment algorithm in clouds." Iainternational Journal for Research in Science and Advance Technology, Issue-1 1 (2018).

[35] Shiri, A., & Khosroshahi, G. K. (2019, April). An FPGA implementation of singular value decomposition. In 2019 27th Iranian Conference on Electrical Engineering (ICEE) (pp. 416-422). IEEE.

[36] Trik, M., Bidgoli, A. M., Vashani, H., & Mozaffari, S. P. (2022). A new adaptive selection strategy for reducing latency in networks on chip. *Integration*.