International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 04 | Apr 2023 www.irjet.net p-ISSN: 2395-0072

Optimizing Task Scheduling in Mobile Cloud Computing Using Particle Swarm Optimization (PSO) Algorithm

Shouaib Scander[1], Souheil Khawatmi[2], Yaser Fawaz [3]

[1]pursuing a M. S. System and Computer Networks at the Faculty of Informatics Engineering, University of Aleppo, Syria.
[2]associate professor at the Faculty of Informatics Engineering, University of Aleppo, Syria.
[3]associate professor at the Faculty of Informatics Engineering, University of Aleppo, Syria.

---***---

Abstract - *Cloud computing is a modern type of shared infrastructure that could interconnect large groups of systems and allows user to connect via the Internet. The term cloud is an expression used to refer to the internet and The Internet is the basis on which cloud computing depends. One of the most necessary requirements in cloud computing system is task scheduling, which plays a main role in the performance of each part of cloud computing equipment. Task scheduling determines tasks that need to be sent to the appropriate virtual device to meet user-defined quality of service (QoS) constraints such as completion time and cost in cloud. In our paper, we have proposed a comprehensive multi-purpose task scheduling optimization model which reduces task transmitting time, execution time and cost. The proposed model is built based on Particle Swarm Optimization (PSO), and the implementation results offer that the new proposed model is more dynamic in speeding up tasks execution and decreasing costs.*

Key Words: Offloading, Mobile Cloud Computing, Tasks Scheduling, Particle Swarm Optimization (PSO)

1.INTRODUCTION

Cloud computing is a new type of computing that provides applications, data and all computer services dynamically over the internet [1]. Now it has become one of the most important fields of information technology. Organizations that need additional resources to develop their data centers rent these resources from the cloud computing system instead of purchasing those resources and pay according to their use of these additional resources [2].

Many applications require powerful computing devices and consume too much energy. Therefore, it is not a good idea to run such applications on constrained-resources devices, such as mobile devices, since they have limited computing power and battery life [3]. To handle this situation, researchers proposed tasks offloading to run the resource-hungry applications on the cloud [4].

Offloading is a technology by which large applications or parts of them i. e., tasks on local devices, are sent to the cloud to run there, as shown in Figure 1. Then, the results are sent back to the end devices, thus, the offloading process reduces the execution time and power consumption of any task run on mobile devices [5].

Although cloud computing provides many services, there are many problems with it. one of these problems, scheduling tasks that is one of the most critical problems due to the need to establish a suitable sequence to divide these tasks [6]. Therefore, cloud computing uses scheduling algorithms to assign confirmed tasks to nominated resources at an accurate time [7], mostly focused on cloud performance optimization which is bandwidth, memory and time discount.

Tasks scheduling are split into two parts: one is used as a unified scheduler for the resource gathering, fundamentally responsible for scheduling cloud APIs and applications, and the other one is for scheduling unified port resources in the cloud such as task scheduler. [8].In this field, some researchers have published papers on the problem of scheduling tasks to get better performance using most optimization methods. The aim of these papers was to reduce cost, response time, uptime and resource usage [9].

The rest of this paper is organized as follows: section II presents Research motivation. section III presents Background and related works. section IV presents Mathematical models. section V presents Implementation and performance evaluation. section VI Conclusion and future work.

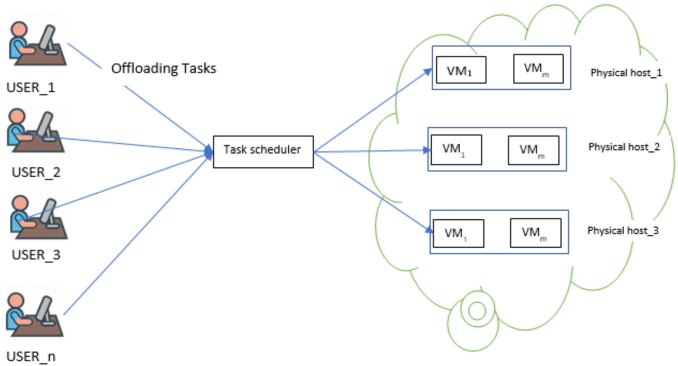

Figure 1: Task scheduling-based cloud scheme

2. Research motivation

In cloud computing, there are a large number of virtual devices that have different processing and storage capabilities. Therefore, the cloud usage depends on the ability of the virtual machines to process, store, transmit the data, taking into account the maintenance of hardware and cloud resources consumption [6].

Optimizing performance and resource usage has an inverse relationship i.e., increasing resource usage leads to waiting time and thus reducing device performance on the cloud [9].

Therefore, to achieve the maximum use of resources and without affecting performance, resources should be allocated efficiently Simultaneously to achieve common goals between consumers and service providers [10].

As shown in Figure 2, there are some of tasks with number (n) and a group of virtual devices (m). Any task can be carried out by any virtual devices. In our paper, we try to find the best solution for distributing tasks on virtual devices in order to reduce the execution time and cost of the task.

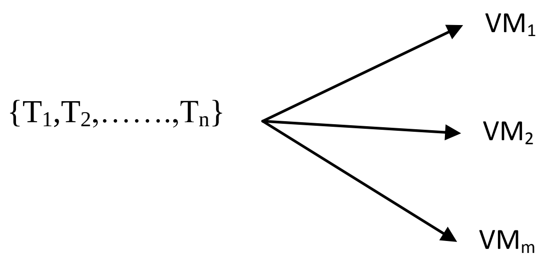

Figure 2: Assigning tasks to virtual machines

3. Background and related works

3.1 Task scheduling algorithms in the cloud:

As shown in TABLE I, task scheduling algorithms are classified into the two categories:

1. Static scheduling algorithms: Algorithms of this type are considered simple compared to dynamic algorithms. They depend on the previous known information of the system, and they do not consider the current state of devices on the cloud server, as they only divide tasks on virtual devices in the cloud server [11].

2. Dynamic scheduling algorithms: these algorithms do not consider the system's previous known information but rather the virtual devices current state. Tasks are distributed to the virtual machines according to their current computational ability [11].

3.2. Particle swarm optimization algorithm:

It is one of the dynamic scheduling algorithms, and swarm intelligence is one of the branches of artificial intelligence that studies the collective behavior of several components, widely used in solving optimization problems [15].

The algorithm of Particle Swarm Optimization belongs to the class of swarm intelligence techniques, in which obtaining the optimal solution is done by simulating the behavior of birds while searching for food [16]. Therefore, any system that depends on this algorithm will initially be formed from a random pool of random solutions. Then, the algorithm searches for the optimal solution within this assembly.

In our model, we call every solution a " bird" in the search field "particle", and every "particle" have "fitness" values that are calculated using "fitness function" in order to reach the optimal solution.

The algorithm, as illustrated in Figure 3, is initialized with a set of random birds (solutions). After that, the optimal solution is searched by updating the generations. In each cycle, each bird is updated depending on the following variables:

1. The best "Fitness" this bird has reached, which is always be stored as "Pbest".

2. The best fitness this swarm has reached, called as "Gbest".

3. The value of the best local positioning of this bird compared to the neighboring birds, called as "Ibest".

After selecting the best value of " Gbest ", the bird changes its speed and position following these two equations:

$$V[] = v[]+C_1*rand()*(Pbest[]-present[]) +C_2*rand()*(Gbest[]-present[\,]) \quad (1)$$

$$Present[\,] = present[\,] + v[\,] \quad (2)$$

Where V[ ] is the speed of the bird, Pbest [ ] and Gbest [ ] are random numbers between (0 and 1), C_1 and C_2 are learning factors (generally $C_1 = C_2 = 2$), and Present[ ] is the current bird.

TABLE I: A COMPARISON OF TASKS SCHEDULING ALGORITHMS

Algorithms	Methodology	Advantages	Disadvantages
GA[12]	It is one of the algorithms that depends on a set of randomly generated solutions	The algorithm reduces the search space.	The algorithm consumes a good amount of time when the search space is relatively large because it is executing a large number of iterations
ACO [13]	It is one of the Swarm Intelligence, A technique for processing computational tasks by searching the shortest path	The ant colony algorithm is basically a distributed parallel algorithm and it is powerful and easy to integrate with other methods.	Ant colony algorithm has slow convergence speed, long computation time
FCFS (first come first serve) [14]	The order of tasks in the algorithm depends on the network latency.	• Simple and popular task scheduling algorithm • More justice than other scheduling algorithms • It depends on FIFO.	• The waiting time for tasks is relatively large. • It does not give importance to any task, i.e., if at the beginning of the queue there is a large task, all tasks will wait a long time. • It does not utilize resources in an optimal way.
SJF (shortest job first) [14]	Tasks are arranged according to their priority, where priority is set depending on the length of the task (small task = high priority).	Average waiting time is less than other algorithms.	Large turnaround time for large tasks, and the worst case is that the waiting time for a large task is infinite.
MAX-MIN [14]	The tasks are classified according to the time required to execute, as the tasks that require a considerable execution time are executed first.	optimal investigation of resources and The performance of the MAX-MIN algorithm is better than last algorithms.	Waiting time for small tasks is relatively large.

4. Mathematical models:

First, we will start by defining a set of symbols necessary to build a mathematical model for scheduling tasks, as follows:

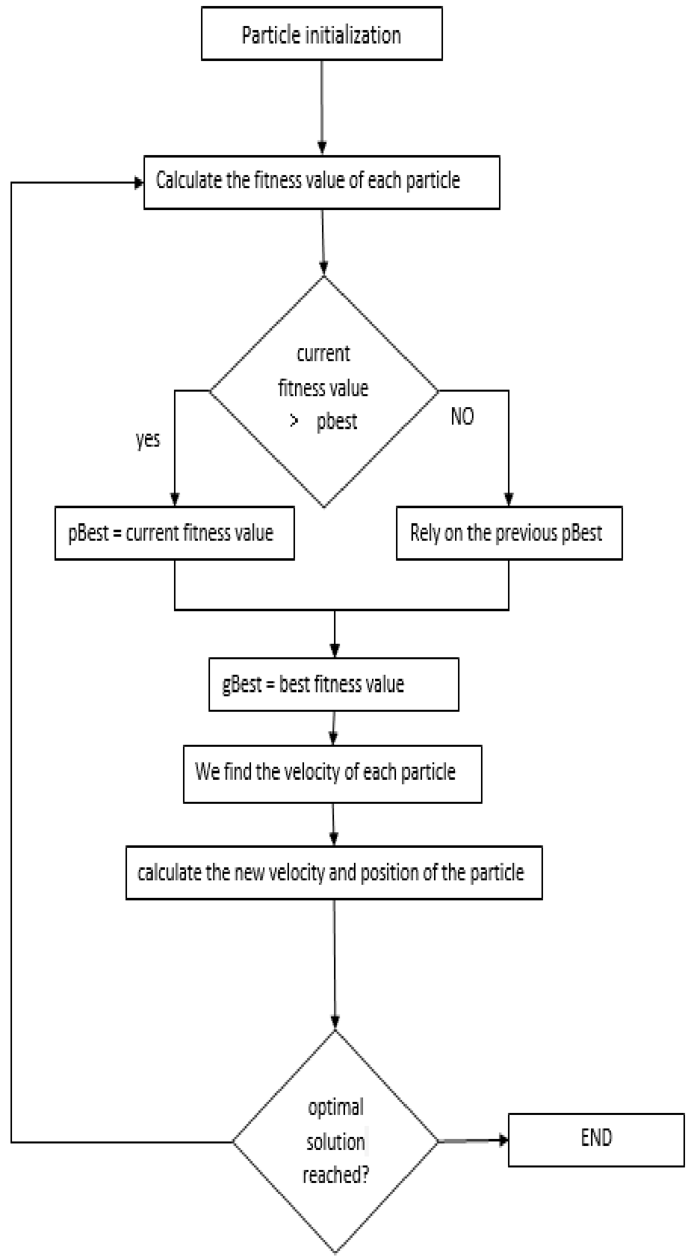

Figure 3: Flowchart of the algorithm PSO

1. n represents the number of tasks will be processed

2. T = {T_1,T_2,...,T_n } represents task "t " with index " i " where " i " between {1} and{n} .

3. m represents the number of VM, and VM = {VM_1, VM_2,..., VM_m) are the available virtual machines.

4. SVM_k represents the capacity of the virtual machine with index k where $1 \leq k \leq m$

5. BW_{ik} is the bandwidth between the task number (i) and the virtual machine number (k).

6. X_{ik} is used to assign a task i to a virtual device k, i.e., X_{ik} = 1 means task i is sent to virtual device while X_{ik} =0 means task i is executed locally.

7. W_{ti} weight of a task i which needs to be offloaded to a virtual device.

8. $D_{T_{ik}}$ is the amount of data exchanged between task i, which created this data, and the VMk that will execute the task.

9. T_{send} is the time needed to send data to the cloud server.

10. $T_{exe.}$ Is the time required for execution on the server in the cloud.

The execution time we need to assign tasks to virtual devices can be calculated using the equation (3):

$$T_{exe} = \sum_{i=1}^{n} \sum_{k=1}^{m} X_{ik} * \frac{W_{ti}}{SVM_k} \quad (3)$$

Send Time, which is the time for the transmitting the method from mobile device to the cloud and back to the mobile device, is calculated using the equation (4):

$$T_{send} = 2 * \sum_{i=1}^{n} \sum_{k=1}^{m} X_{ik} * \frac{D_{T_{ik}}}{BW_{ik}} \quad (4)$$

The total time can be calculated by adding the Texe to the Tsend using the equation (5):

$$Total\_time = T_{exe} + T_{send} \quad (5)$$

5. Implementation and performance evaluation:

In order to test our proposed model, we used the following settings:

- The length of tasks is set to be between 50 and 200 Million Instructions Per Second (MIPS).
- The capacity of the virtual machine is between 2.0 and 4.0 MIPS.
- Transmitted data size varies from 50 to 150 Kb/sec.
- Bandwidth varies between 25 and 50 kb/s.

We used a PC, which has CPU Intel (R) Core (TM) i3 with 4GB RAM to test the algorithm under Microsoft windows OS. The experiment carried out on the CloudSim simulator within the NetBeans environment [17].

For 5 physical devices, there are 16 virtual devices were randomly distributed to assign tasks to virtual machines. The results were as follows:

As shown in Figure 4, the execution time using swarm scheduling took less time comparing with random scheduling, and the improvement was near to 31%.

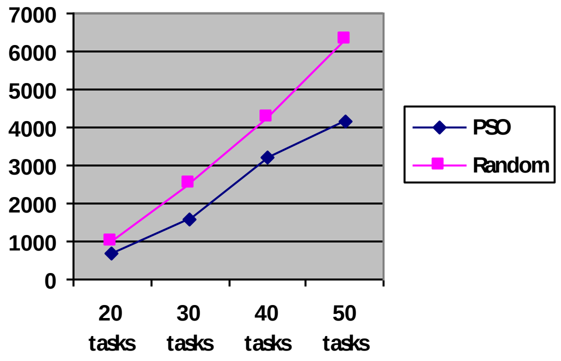

Figure 4: Execution time

As shown in Figure 5, the running task Cost using swarm scheduling was also less comparing random scheduling, and the improvement was near to 30%.

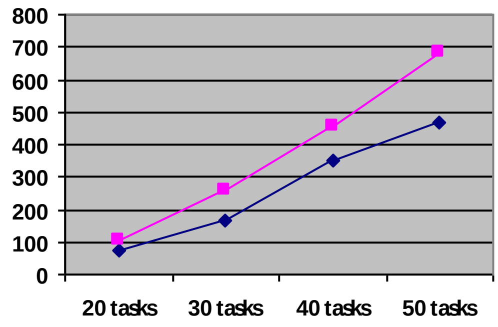

Figure 5: Processing and transmission data cost

6. CONCLUSIONS and Future Work

The process offloading tasks from mobile devices to the cloud led to an increase in the load on the servers in the cloud, which also led to the delay in executing tasks, so a scheduling algorithm was worked out in order to reduce the load.

In this article, we suggested using the particle swarm optimization algorithm, which has greatly improved the optimal distribution of tasks on virtual machines.

The main idea of this paper is to achieve efficiency in tasks scheduling on the cloud. We used a cloud simulator to implement the proposed algorithm and presented the results, which showed a significant improvement in balancing the distribution of tasks on virtual devices in the cloud.

As a future work, we are going to use the genetic algorithm to balance the load of virtual machines on the physical machines. Thus, the output of the genetic algorithm will be used as an input to the particle swarm optimization algorithm to balance the distribution of tasks on available virtual devices.

REFERENCES

[1] Biswas, Milon, and M. D. Whaiduzzaman. "Efficient mobile cloud computing through computation offloading." Int. J. Adv. Technol 10.2 (2018): 32.

[2] Hao, Jialu & Xian, Ming & Wang, Huimei & Tang, Fengyi & Xiao, Pu. (2018). Mobile Cloud Computing: the State of Art, Application Scenarios and Challenges. 1-5. 10.1109/CIACT.2018.8480365.

[3] Oliveira, Rômulo Reis de et al. "Improving the Trade-Off between Performance and Energy Saving in Mobile Devices through a Transparent Code Offloading Technique." (2019).

[4] Jin, Xiaomin, et al. "A survey of research on computation offloading in mobile cloud computing." Wireless Networks (2022): 1-23.

[5] Raji, Babatunde O., and Olusola O. Ajayi. "Implementing a Mobile Cloud Computing Framework with an Optimized Computational Offloading Algorithm." (2019).

[6] Ramotra, Atul Kumar, and Anju Guide Bala. Task-Aware Priority Based Scheduling in Cloud Computing. Diss. 2013.

[7] B. A. Al-Maytami, P. Fan, A. Hussain, T. Baker and P. Liatsis, "A Task Scheduling Algorithm With Improved Makespan Based on Prediction of Tasks Computation Time algorithm for Cloud Computing," in IEEE Access, vol. 7, pp. 160916-160926, 2019, doi: 10.1109/ACCESS.2019.2948704.

[8] Volkova V. N., Chernenkaya L. V., Desyatirikova E. N., Hajali Moussa, Khodar Almothana, Alkaadi Osama ―Load Balancing in Cloud Computing‖ in Proc. Of 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (2018 ElConRus),

St. Petersburg and Moscow, Russia, January 29 - February1,2018,pp.397400\.doi:10.1109/EIConRus.2018.8317113.

[9] S. Shukla, A. K. Singh and V. Kumar Sharma, "Survey on Importance of Load Balancing for Cloud Computing," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021, pp. 1479-1484, doi: 10.1109/ICAC3N53548.2021.9725442.

[10] Kumar Garg S, Buyya R. Green Cloud Computing and Environmental Sustainability, Australia: Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering, The University of Melbourne; 2012

[11] Ibrahim, Ibrahim Mahmood. "Task scheduling algorithms in cloud computing: A review." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12.4 (2021): 1041-1053.

[12] D. Subramoney and C. N. Nyirenda, "A Comparative Evaluation of Population-based Optimization Algorithms for Workflow Scheduling in Cloud-Fog Environments," 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 2020, pp. 760-767, doi: 10.1109/SSCI47803.2020.9308549.

[13] Yamini, R., and M. Germanus Alex. "Comparison of resource optimization algorithms in cloud computing." International Journal of Pure and Applied Mathematics 116, no. 21 (2017): 847-855.

[14] Aladwani, Tahani. "Types of task scheduling algorithms in cloud computing environment." Scheduling Problems-New Applications and Trends (2020).

[15] A. Khodar, L. V. Chernenkaya, I. Alkhayat, H. A. Fadhil Al-Afare and E. N. Desyatirikova, "Design Model to Improve Task Scheduling in Cloud Computing Based on Particle Swarm Optimization," 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2020, pp. 345-350, doi: 10.1109/EIConRus49466.2020.9039501.

[16] Al-Olimat, Hussein S., et al. "Cloudlet scheduling with particle swarm optimization." Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on. IEEE, 2015.

[17] Dejan Bulaja, Kristina Božić, Nikola Penevski Nebojša Bačanin Džakula."INTRODUCTION TO CLOUDSIM",2019.

BIOGRAPHIES

Shouaib scander received his B. S. degree from the Faculty of Informatics Engineering, Tishreen University, Syria in 2018. He is currently pursuing a M. S. Computer Networks at the Faculty of Informatics Engineering, University of Aleppo, Syria. His current research interests include mobile cloud computing.

Dr. Souheil Khawatmi received his B. S. degree from the Faculty of Computer Engineering, University of Aleppo, Syria

in 1982 and PhD degrees from the Institute of saint-Petersburg for Communication, Germany in 1989. He is currently an associate professor at the Faculty of Informatics Engineering, University of Aleppo, Syria. His current research focuses on Computer Networks and Wireless Communication

Dr. Yaser Fawaz
He received his PhD from INSA of Lyon - France.
Currently, he is the head of systems and computer networks at the faculty of informatics Engineering. University of Aleppo