Optimizing Task Scheduling Using Service Quality Measures and Data-aware Scheduler in Cloud Computing Environment

Ensieh Azmayandeh , **Mohammadreza Farahpour Haghani,** **Elham jafari,**

Sama technical and vocatinal training college Islamic Azad University, Rasht Branch, Rasht, Iran
Social Security organization, Rasht, Iran
Sama technical and vocatinal training college Islamic Azad University, Rasht Branch, Rasht, Iran

Summary
Cloud computing can provide flexible resources and dynamic and scalable services through the internet. Considering these dynamic resources, their allocation to tasks is one of the most challenging issues in the cloud computing environment. Accordingly, different scheduling algorithms have been proposed for cloud computing. These algorithms investigate different factors, including the specified service quality by users, execution time, cost, load balancing and energy consumption in task execution, fairness, and utility of resources. This paper aims to propose a method to minimize the completion time and cost using the ant colony algorithm and data-aware scheduler. More specifically, first tasks are categorized into two groups based on completion time and cost, the optimal path is found using the ant colony algorithm, and the most appropriate virtual machine is selected using the data-aware scheduler technique based on bandwidth, the load balancing of the source machine, and the load balancing of the destination machine. Results are presented using cloudsim.
Keywords:
Cloud computing, Task scheduling, Service quality, Ant colony, Data-aware scheduler.

1. Introduction

Cloud computing provides a virtual set of resources to its users through the internet [1]. Since ability and accessibility are different for dynamic resources, task scheduling is considered one of the most challenging issues in cloud computing. The reason is that scheduling aims to optimally allocate tasks to resources at optimal time by an appropriate approach. Accordingly, there have been different scheduling algorithms in cloud computing. These algorithms investigate different factors, including the quality of service determined by the user, execution time, cost, load balancing, and energy consumption in task execution, fairness, and utility of resources. In all algorithms, it is difficult to find a proper solution, which not only reduces the total completion time of tasks and improve the efficiency of processors, but also insures load balancing among resources [2].

This paper proposes an optimal algorithm to insure completion time and cost measures by the service quality parameter. Accordingly, the tasks are divided into two groups based on completion time and cost. The ant colony algorithm then selects the optimal path and allocates the optimal virtual machine to the tasks using a data-aware scheduler. The rest of this paper is organized as follows. Section 2 presents a brief review of cloud computing and task scheduling. Sections 3 and 4 discuss the quality of the ant colony algorithm and data-aware scheduler. Sections 5, 6, and 7 present the structure of the proposed method, evaluation, and analysis of results using Cloudsim.

2. REVIEW OF CLOUD COMPUTING AND TASK SCHEDULING

Cloud computing systems provide its users with online commercial applications through web browsers or other software applications. Information and software applications are stored on servers and provided to users on demand. In this architecture, the data is mostly located on online servers and applications are executed on cloud servers, as well as users` browsers [3].

The overall architecture of cloud computing includes the 5 user layers, software as a service (SaaS), platform as a service (PaaS), infrastructure as a service (IaaS), and server layer [4]. However, users may mostly experience the presence and progress of the first and second layer; whereas, other layers are also very important for service providers and developers [5].

Task scheduling is one of the key processes in the service infrastructure layer, which aims to efficiently execute the received requests on the resources by considering other features of the cloud environment [6]. Scheduling is the process of mapping tasks to available resources based on their features and requirements. Available resources should be effectively used regardless of cloud service parameters [7]. Cloud scheduling can be divided into the tree stages of resource discovery, information collection, and task execution [8].

Scheduling aims to insure minimum execution time, maximum processor efficiency, and load balancing among all or a combination of processers. Accordingly,

Manuscript received April 5, 2025
Manuscript revised April 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.4.18

different approaches have been proposed. Most of these algorithms disregard measures like utilization rate, resource usage, load balancing, and the necessity of rapid response. Therefore, task scheduling is one of the important issues in cloud computing systems. The reason is that a scheduling algorithm is proposed to create an optimal scheduler, such that tasks are allocated to the most appropriate resources at optimal time.

There are different algorithms for task scheduling. Most of these algorithms disregard measures like utilization rate, resource usage, load balancing, and the necessity of rapid response [9]. Whereas, in addition to these measures, user requirements should also be taken into account. Therefore, an algorithm is appropriate, which provides user requirements, i.e. service quality parameters [10].

Therefore, based on the diversity of users, in order to allocate a task to the most appropriate resource, service quality parameters should be investigated, including the network bandwidth, service completion time, service reliability, and costs [11]. User satisfaction is measured based on these parameters, which is considered the quantitative evaluation of different standards.

Moreover, task scheduling in the cloud is a NP-Hard problem due to several reasons, including dynamic and inhomogeneous characteristics of resources and requests in the cloud computingenvironment [12]. Accordingly, heuristic methods should be used to solve such problems. Among the proposed heuristic approaches, the ant colony algorithm is more appropriate for NP-Hard problems due to its simple implementation and low cost in comparison to other methods [13]. Therefore, what follows explains the ant colony algorithm to propose an optimal task scheduling algorithm.

3. THE ANT COLONY ALGORITHM

The ant colony algorithm is one of the most efficient algorithms used to solve hybrid optimization problems in inhomogeneous and dynamic environments. The main notion of the ant colony algorithm is as follows [14].

- At time 0, each ant randomly selects a node. At this stage, the initial pheromone value $\tau_{ij}(0)$ is computed for edges (i, j).
- Ants select the next node based on probability equation (1) .This stage is repeated until all nodes are checked.

$$\rho_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{k\epsilon\, allowed_k}[\tau_{ik}(t)]^{\alpha}[\eta_{ik}(t)]^{\beta}} & \text{If} \quad k\epsilon\, allowed_k \\ 0 & \text{Otherwise} \end{cases} \qquad (1)$$

Where, $\tau_{ij}(t)$ is the feromon value on edge (i, j) and $\eta_{ij}(t)$ is the heuristic information value, equal to $\eta_{ij}(t) = \frac{1}{d_{ij}}$. Moreover, d_{ij} is the distance between nodes i and j. finally, αandβ are two parameters to control the relative feromon weight and heuristic information value.

- Update the pheromone of each node
- Compare pheromone values and select the most optimal and effective path [15].

Inspired by the ant colony algorithm, the proposed algorithm considers the computational capacity of a resource as pheromones. However, both processor and storage resources should be scheduled, since the cloud computing environment is dynamic and posed with a large volume of requests. In dynamic and distributed environments, the data-aware scheduler is an appropriate technique to schedule the storage resources. Therefore, in the cloud computing environment, the data-aware scheduler is used to calculate the selection probability of the next resource.

4. THE DATA-AWARE SCHEDULER

The data-aware scheduler is proposed for dynamic and distributed environments. The data-aware scheduler recovers information about available resources and load information of each node. Based on this information, the data-aware scheduler module makes decisions about selecting an appropriate node to schedule different tasks [16].

This scheduler is also designed for data placement and transfer, which takes into account scheduling features like data transmission, storage location, freeing, and separating data. This scheduler allows users to schedule both the processor and storage resources and prevent creating long waiting queues to allocate tasks to virtual machines [17].

In the data-aware scheduler, whenever there is a large volume of tasks, they should be separated, independent, and the required data replicate should be stored in the same node. However, a replica management mechanism is also added since the number of data replica is not sufficient at this state. In the replica management mechanism, data replica should be change after scheduling. It means that the scheduler is responsible for the replica generated during its own scheduling process. The replica management mechanism includes the following stages [16]:

1- The time of generating a new data replicate
2- A method to select a new node to store the data replicate
3- Pre scheduling dependent tasks
4- The elapsed time of the data replicate

Three effective factors to select a new node are the network bandwidth and the load balancing of the source and destination nodes [16].

5. THE PROPOSED METHOD

The proposed method groups user tasks based on the two minimum cost and minimum service completion time measures of the service quality parameters. This reduces the communication cost. Moreover, grouping also determine the priority of running tasks. More specifically, in the task group based on the service completion time, the tasks must be executed first, which have the least service completion time. Tasks are sorted in ascending order based on the least completion time. The workflow time of each resource is calculated as equation (2).

Cost= virtual machine cost * (task length / virtual machine processing power) (2)

After computing the workflow time of each virtual machine, the virtual machine with the least workflow time is selected to run the task. Tasks are prioritized based on their cost as equation (3).

Cost= virtual machine cost * (task length / virtual machine processing power) (3)

Accordingly, the virtual machine with the minimum cost is selected to schedule tasks as long as it has free capacity. Therefore, the received tasks are grouped based on their completion time and cost. Based on the priority of each task in each group, tasks are then selected and allocated to the optimal virtual machine obtained by the ant colony algorithm. Generally, the proposed algorithm has the following process.

There is a set of tasks $T=\{T_1, T_2, \ldots, T_m\}$, where m is the total number of tasks and a set of virtual machines $vm=\{vm_1, vm_2, \ldots, vm_n\}$, where n is the total number of virtual machines.

Stage 1: received tasks are sent to the broker based on their completion time and cost.

Stage 2: after grouping tasks, based on the ant colony algorithm decision making to select resources and the data-aware scheduler to calculate the pheromone, the proposed scheduling scheme is as follows.

Stage 3: The initial pheromone value is computed for each virtual machine. More specifically, ants are first randomly placed on virtual machines for the first task. Here, the pheromone represents the total workflow, completion cost, or resource suitability. The pheromone of VM_j is computed as equation (4).

$T_j(0)$=TA or Cost + $suitable\_node\_dataaware_j$ (4)

Where, num_proc_j is the number of processors and $suitable\_node\_dataaware_j$ is the suitability of VM_j, which is calculated as equation (5) using the data-aware scheduler technique.

$suitable\_node\_dataaware_j$ = network bandwidth + load balancing of the source node + load balancing of the destination node (5)

Stage 4: The decision making mechanism and virtual machine selection based on the completion time

A.1. K ants select a task based on its priority from the task list and allocate it to VM_j based on equation (6). The decision making mechanism is based on the workflow time.

$$\rho_j^k = \begin{cases} \dfrac{[\tau_j(t)]^\alpha [TA_j]^\beta}{\sum[\tau(t)]^\alpha [TA]^\beta} & \text{If} \quad j \in 1 \ldots n \\ 0 & \text{Otherwise} \end{cases} \quad (6)$$

Where, $\tau_j(t)$ is the pheromone at time t and TA is the workflow time of VM_j.

A.2. When an ant finds an optimal solution for each task, the pheromone of the selected virtual machine is updated by the best ant with the least completion time. Since, $\tau_j(t)$ represents the pheromone intensity of VM_j at time t, pheromone is updated by equation (7).

$$\tau_j(t+1) = (1-\rho) * \tau_j(t) + \Delta\tau_j \quad (7)$$

Where, $\rho \epsilon (0,1]$ is the pheromone reduction coefficient. The highest value is ρ and the lowest value is equal to the pheromone of the last solution. Moreover, $\Delta\tau_j$ is equal to $\Delta\tau_{j} = 1/T_{ik}$ and T_{ik} shows the shortest path searched by the kth ant in the its iteration. Finally, the task is allocated to the best and most appropriate virtual machine to minimize the workflow time, which also minimizes the completion time.

B. The decision making mechanism and virtual machine selection based on cost:

B.1. K ants select a task based on its priority from the task list and allocate it to VM_j based on equation (8). The decision making mechanism is based on the cost.

$$p_j^k = \begin{cases} \dfrac{[\tau_j(t)]^{\alpha}[cost_j]^{\beta}}{\sum[\tau(t)]^{\alpha}[cost]^{\beta}} & \text{If} \quad j \in 1 \ldots n \\ 0 & \text{Otherwise} \end{cases} \quad (8)$$

B.2. when an ant finds an optimal solution for each task, the pheromone of the selected virtual machine is updated by the best ant with the least completion time. Since, $\tau_j(t)$ represents the pheromone intensity of VM_j at time t, pheromone is updated by equation (7). The rest of the computations are similar to the previous group. Finally, the virtual machine is selected with the least cost and tasks are scheduled on it as long as it has free capacity.

6. Simulation Results

The proposed algorithm is simulated using cloudsim software in the Linux operating system. Simulated algorithms include the proposed and FCFS algorithms (the default scheduling scheme of cloudsim).

The proposed algorithm aims to minimize the completion time and cost to insure user satisfaction. Therefore, the proposed algorithm evaluates the minimum task completion time and cost among all virtual machines. The simulation environment is implemented in cloudsim as follows.

First, a data center is created and then the data center provides a set of resources, e.g. CPU, memory, bandwidth, etc. subsequently, the Datacenter Broker is created to simulate a broker. A broker is an interface or delegator, which is the interface between the software as a service layer and the cloud provider. Broker operations are mostly based at this layer, which can find an appropriate cloud provider to allocate resources and services. This class sends requests to data centers to allocate virtual machines. Table 1 presents the configured parameters of the cloud simulator.

TABLE 1. The configured parameters of the simulator

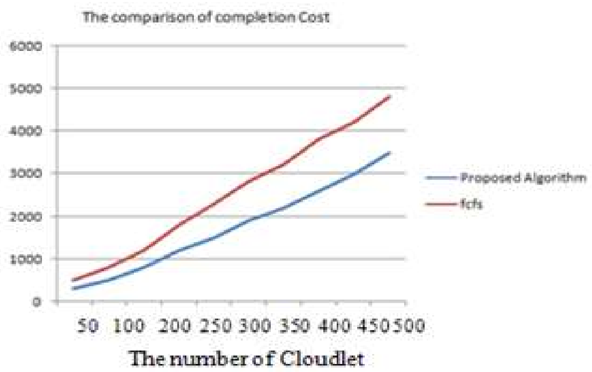
Type	Parameters	Count
Data Center Broker	number of data centers	10
	Number of hosts	2-6
virtual machine (VM)	Number of vms	50
	RAM	512-2048 MB
	Bandwidth	500-1000
Tasks or cloudlet	Total number of tasks	100-500

The parameters of the simulated algorithm are as follows in table 2.

TABLE 1. THE CONFIGURED PARAMETERS OF THE SIMULATOR

Parameters	Count
Number of tasks	100-500
Number of iterations	50
α	0.5
β	0.5

Parameters α and β are used to control the relative weight between the pheromone, the value of workflow time, and cost, whose sum should be equal to one.
The simulation is repeated for different cloudlets (tasks) using the proposed and FCFS algorithms (default cloudsim scheduling). Fig.1 presents the results according to equations (2) and (3).



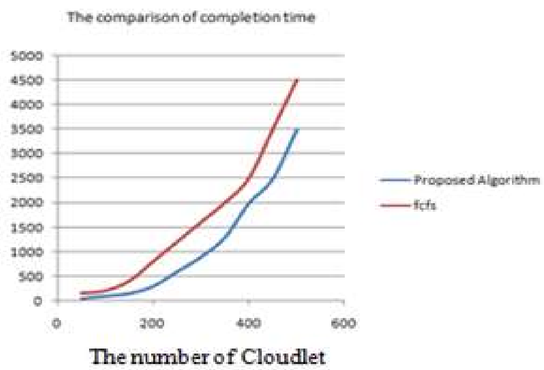


Fig. 1. Simulation of the proposed algorithm, FCFS, and the comparison of completion time and cost

As we can see, in comparison to FCFS, the proposed algorithm minimizes the completion time and cost. Moreover, it seems that by increasing the number of tasks, the proposed algorithm has a lower execution time than that of FCFS. The reason is that the ant colony algorithm and data-aware scheduler select the optimal virtual machine through pheromones and using the completion time and cost parameters and the proposed method has a better performance with a higher number of tasks.

7. Conclusions

Due to dynamic resources of cloud computing, task scheduling is its most challenging issue. Accordingly, this paper introduces the cloud computing technology and discusses its scheduling issue. An appropriate scheduling method can allocate tasks to machines with minimum cost and time. therefore, an approach was proposed to first group tasks and then select an optimal virtual machine using the ant colony algorithm and data-aware scheduler. Using cloudsim, results were evaluated and it was shown that the proposed method minimizes the execution time, cost, and completion time of the tasks.

Reference

[1] Rajkumar Buyya, chee shin yeo, Srikumar Venugopal, James Broberg, Ivona Brand, cloud computing and emerging IT platform: Vision, hypr, and reality for delivering computing as 5th utility, *Future Generation Computer System, ELSEVEIR*, (2008).

[2] Paul, M., Sanyal, G., Survey and analysis of optimal scheduling strategies in cloud environment, *IEEE*,(2012).

[3] Junwei Ge, Bo Zhang, and Yiqui Fang, "Research on the Resource Monitoring Model Under Cloud Computing Envirment", *Springer*, 2010.

[4] Christian Baun, Jens Nimis, Stefan Tai. Cloud Computing. *London : Springer* , 978-3-642-20916-1, (2011).

[5] Jean-Paul Ebejer, Simone Fulle , Garrett M. Morris, Paul W. Finn, The emerging role of cloud computing in molecular modelling. Oxford Centre for Innovation, *New Road : Journal of Molecular Graphics and Modelling*, Elsevier Inc,Vol. 44, (2013).

[6] R.Madhubala, An Illustrative study on Cloud Computing, *International Journal of Soft Computing and Engineering*, vol. 1, issue. 6, (January 2012).

[7] Shi, Daji Ergu · Gang Kou · Yi Peng · Yong Shi · Yu. ,The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment, *Springer Science+Business Media*, DOI 10.1007/s11227-011-0625-1, (2013).

[8] Fatos Xhafa, Ajith Abraham, Metaheuristics for Scheduling in Distributed Computing Environments. *Spain : Studies in Computational Intelligence,Springer-Verlag Berlin Heidelberg*, 978-3-540-69260-7, (2008).

[9] Tayal, Sandeep, Tasks Scheduling Optimization for the cloud computing Systems, *India : (IJAEST) International Journal Of Advanced Engineering Sciences And Technologies*, Vol. VI, ISBN:2230-7818, (2011).

[10] Sindhu, S., Mukherjee S, Efficient Task Scheduling Algorithms for Cloud Computing Environment , *High Performance Architecture and Grid Computing. Communications in Computer and Information Science*, vol 169. Springer, Berlin, Heidelberg, (2011).

[11] HUANG Qi-yi, HUANG Ting-lei, An Optimistic Job Scheduling Strategy based on QoS for Cloud Computing, *IEEE*, 978-1-4244-6837-9/673-676, (2010).

[12] Y.J. Feng, L. Yu, G.L. Zhang ,Ant colony pattern search algorithms for unconstrained and bound constrained optimization, *Applied Mathematics and Computation, Published by Elsevier* , (2007).

[13] Arūnas Andziulis, Dalė Dzemydienė, Raimundas Steponavičius, Sergej Jakovlev, Comparison Of Two Heuristic Approaches For Solving The Production Scheduling Problem, *Information Technology And Control*, Vol. 40,(2011).

[14] Zhang, Zehua Zhang and Xuejie, A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation. *china : 2010 2nd International Conference on Industrial Mechatronics and Automation, IEEE*, Vol. 10, ISBN: 978-1-4244-7656-5,(2010).