How to Cite:
Gandhi, S. Y., & Revathi, T. (2022). An improved hybrid cloud workflow scheduling algorithm based on ant colony optimization. *International Journal of Health Sciences*, *6*(S4), 869–882. https://doi.org/10.53730/ijhs.v6nS4.5781

An improved hybrid cloud workflow scheduling algorithm based on ant colony optimization

S. Yuvaraj Gandhi
Assistant professor, Department of Computer Science, PSG College of Arts and Science, Coimbatore, Tamil Nadu, India

Dr. T. Revathi
Associate Professor, Department of Computer Science, PSG College of Arts and Science, Coimbatore, Tamil Nadu, India

***Abstract*---**At present, with the enormous development of cloud-based applications, the demand of related enterprises for computing and a large number of storage resources is increasing day by day. Due to the characteristics of ultra-large scale and low cost, cloud computing has been gradually applied to complex workflow scheduling problems in various fields. The traditional scheduling algorithms have lot of limitations such as inefficient task management and unreasonable resource allocation. The workflow scheduling problem in the cloud computing environment has been proved to be an NP-complete problem. To solve these problems, this paper proposes an improved ant colony optimization workflow scheduling for workflow execution of tasks within a user-specified deadline. On this basis, a hybrid cloud deadline-constrained cost workflows scheduling algorithm under the hybrid cloud is proposed, which will prioritize scheduling execution in the private cloud, and use the public cloud to schedule part of the workflow when the task execution time exceeds the task deadline constraint. The simulation results show that the proposed workflow scheduling algorithm reduces the cost and attain faster execution time.

***Keywords*---**cloud computing, cloud scheduling, improved ant colony algorithm, optimization based cloud scheduling.

Introduction

The cloud computing platform uses distributed computing, virtualization and other technologies to integrate a large number of computing equipment and infrastructure, and provides users with computing, data storage and other resources and related services through the network, which is widely used because

Manuscript submitted: 18 Feb 2022, Manuscript revised: 9 March 2022, Accepted for publication: 1 April 2022

of its efficient, cheap and easy to maintain. In addition, it can help companies and enterprises create large data centers with efficient processing power without increasing costs. Hybrid cloud architecture solutions are widely used by major enterprises, hybrid cloud is a hybrid of public and private cloud, enterprises need to use a combination of private cloud and public cloud to take advantage of the unique advantages they provide: that is, after the application workload reaches the threshold level in the private cloud, it suddenly enters the public cloud to obtain additional computing resources, and the public cloud platform saves users time and money [1]. For cloud computing, how you manage cloud computing resources is critical, especially when multiple services are using their applications on the cloud at the same time. Therefore, for resource management, cloud computing requires a suitable scheduling strategy. A good task scheduling algorithm can optimize Quality of Service parameters (QoS), such as maximum completion time, response time, throughput, resource utilization, task rejection rate, reliability, scalability, energy consumption, execution cost, etc., and can consider various constraints, such as deadlines, priorities, economic costs, etc., without violating Service Level Agreements (SLAs), to achieve hard indicator constraints for users, while avoiding load imbalance. The choice of cloud resources in cloud workflow scheduling is a typical NP difficulty [2].

The remaining part of the paper is organised as follows. Section 2 describes about the various literature done in the area of this research and its relevant solutions. Section 3 provides the detailed methodology of the problem. Section 4 provides the pseudo code for the traditional and proposed algorithm and section 5 discusses the results of the problem and section 6 concludes the paper along with cited references.

Literature Survey and Background of the Problem

In the field of cloud computing, scholars at home and abroad have done many valuable researches on workflow scheduling problems and proposed some static scheduling algorithms [3]. Static scheduling algorithms can be divided into heuristic-based algorithms and random search algorithms. According to the main ideas of the user, the heuristic algorithm is divided into: 1) a list class heuristic algorithm that sorts the tasks first according to a certain law, and then dispatches them sequentially; 2) copies the tasks of the parent node to the child nodes, thereby reducing the data transfer consumption; and 3) the cluster class scheduling algorithm that divides the tasks into multiple clusters for scheduling. The random search algorithm mainly refers to the genetic algorithm, bird swarm algorithm and other population algorithms, which use the possible scheduling results as a way for a separate search node to form a search space, and select the scheduling strategy to obtain the results. Depending on the scheduling target, workflow scheduling algorithms can be divided into various types, the most common workflow scheduling goal is how to complete the workflow in the shortest possible time, which is also a common consideration of workflow scheduling in the early grid computing environment. For example, the Earliest Time First (ETF) algorithm in literature [4] selects the earliest time of all the tasks that are ready to be executed each time, and the earliest start time of all the tasks that are ready is selected to schedule, so as to ensure that all tasks have the shortest scheduling time, and the start time of each task is the earliest start time for the task to run on all processors. When two tasks have the same start time, the algorithm selects tasks with a higher static order, ensuring that the next execution node is picked each time. Literature [5] Assigns tasks to the most completed available machines by comparing the earliest time, earliest start time, earliest end time, etc. of the task on different processors.

For task scheduling in a hybrid cloud environment, the document [6] proposes two workflow scheduling algorithms that take into account both finish time and profit for hybrid cloud, the first is the workflow scheduling algorithm DCOH (Deadline-Constrained Cost Optimization for Hybrid Clouds) based on cost optimization in a cost-optimized hybrid cloud based on deadline constraints. A hybrid cloud multi-objective workflow scheduling optimization algorithm MOH (Multi-Objective Optimization for Hybrid Clouds) with the goal of maximum completion time and profit is proposed, and the simulation results show that compared with the existing algorithm, DCOH can reduce the cost of using the public cloud for users, and MOH can provide a good cost and completion time coordination scheme. However, this scheme does not take into account the impact of the change of task execution time, so this paper proposes an algorithm that takes into account the task deadline, through the ant colony algorithm [7, 11]. Ant Colony Optimization (ACO) to optimize workflow scheduling problems in a private cloud environment, the goal of this algorithm is to find the optimal solution for efficient workflow scheduling. This swarm intelligence optimization algorithm is inspired by the behaviour of ants in finding paths in the process of finding food. When the execution time of a task exceeds the set task deadline, the IHCDCW (Improved Hybrid Cloud Deadline-Constrained Cost Workflows Scheduling) algorithm proposed in this article is used to schedule part of the workflow to execute in the public cloud environment.

Hybrid Workflow Scheduling Architecture

In this section, the system architecture of workflows, the task model of workflows, and workflow problem definitions in a hybrid cloud environment are described.

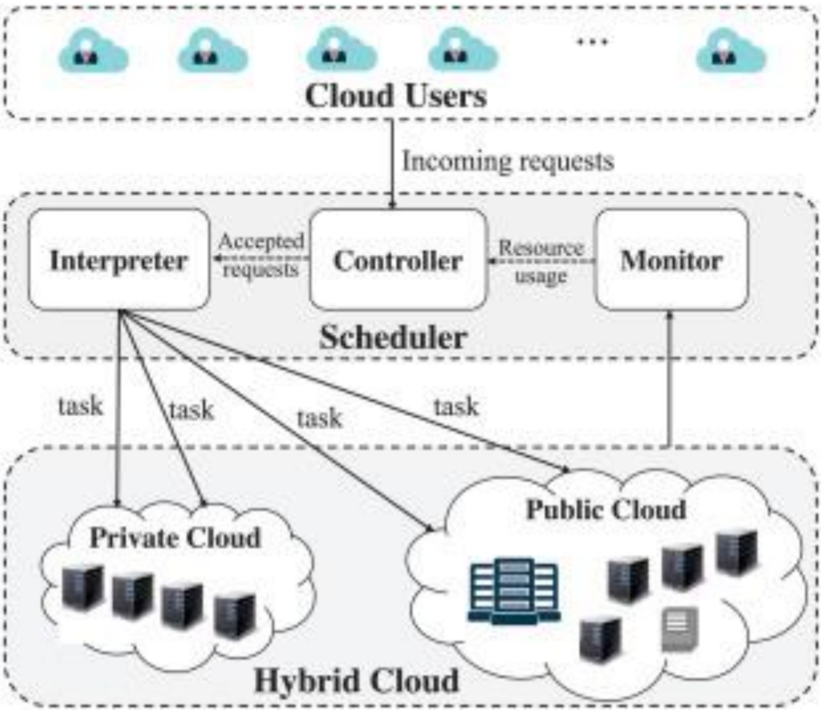

Figure 1. Hybrid cloud workflow system architecture

Hybrid Cloud Workflow System Architecture

Figure 1 shows the workflow system architecture in a hybrid cloud environment [6]. The entire system architecture consists of a workflow scheduling system and hybrid cloud resources. In the system, the workflow scheduling system to receive and judge the task requests initiated by the cloud user, when the cloud user initiated the task continuously arrives, if the request is accepted by the workflow scheduling system, the workflow scheduling system will decide whether the cloud user's request is to use the virtual machine scheduling in the private cloud environment or use the virtual machine that has been rented in the public cloud for scheduling. In the workflow scheduling system, it is mainly divided into four layers, in which the monitor layer (Monitor) is responsible for collecting and sorting out the QoS information of private and public clouds in cloud resources, including the running status, response time, throughput, task rejection rate and other information of cloud resources. The controller is responsible for receiving requests from cloud users, filtering incoming requests when a request arrives, filtering spam requests, and letting the system initiate a compliant workflow schedule. Then the service layer (Service) to handle the workflow scheduling, and according to the cloud resource information transmitted by Mono, the workflow task allocation is carried out, and the proposed algorithm in this paper implements the scheduling allocation at the service layer. Finally, the workflow task is scheduled once by the workflow scheduling algorithm selected in the service layer, and the interaction with the cloud resources is handled by the Manager layer, which does not communicate directly with the virtual machines in the cloud environment, but communicates with the cloud virtual machines through the interface provided by the scheduling Manager layer.

Hybrid Cloud Workflow Task Model

Stand-alone tasks have no dependencies on each other and only need to be specified to the virtual machine by the task scheduler, while workflow tasks have a sequence of execution and are more complex to run. Each workflow task is composed of a series of subtasks and data dependencies between tasks and tasks, there is a data interaction between tasks and tasks, some tasks require the completion of the previous task to continue execution, because of the complexity of this dependency, so use Directed Acyclic Graph (DAG) to describe a workflow task [8]. The DAG diagram used in Figure 2 represents a simple workflow example that includes 9 task nodes and 12 dependencies, with nodes 1 to 9 representing task nodes and edges a-l representing dependencies.

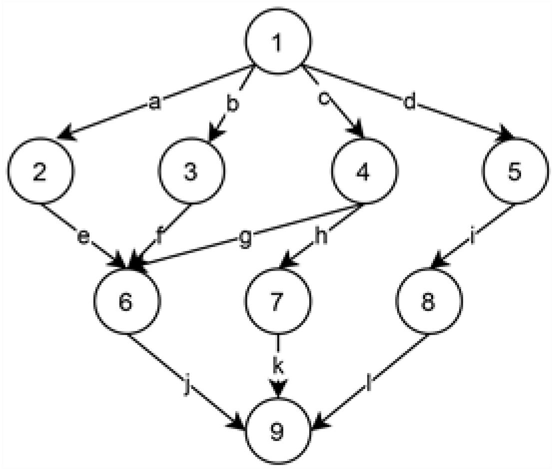

Figure 2. Workflow task model

The usual workflow model can be modelled against a DAG diagram, Represents a workflow task, T represents the subset of subtasks of the workflow, Representing data dependencies between tasks, workflow W is constrained by a deadline, which can be described as a WD (Workflow Deadline) that requires all workflows to complete within this WD deadline. Dependency usage between tasks (ti, tj) means, tj the execution of the task is subject to ti limit, need to get ti Execution of completed data for the workflow to continue execution can be described as , TL (Task Length) represents the length of execution required to perform a workflow task in a virtual machine instance.

Hybrid Cloud Workflow Problem Description

Cloud providers that provide IaaS services to cloud users need to inform users of the parameters of the cloud virtual machine in advance, such as CPU, RAM, disk, and other information. Among them, the virtual machine provided by the cloud service provider can be called an instance, and a virtual machine is called a cloud instance. The quality of CPU computing power in a cloud instance determines how long a task takes to execute in a cloud instance, (t_i, t_j in this model, two tasks

t_i, t_j Data transfers are required between, and the time required depends mainly on the bandwidth between the data transfers. In this article we only consider CPU performance and network communication bandwidth as the main factors for transfer between virtual machine instances. Therefore, cu can be used to represent the computing power of the CPU. Use bw to represent bandwidth.

Definition 1. resource model in a private cloud: a virtual machine instance can be described as equation 1:

$$pr=\{cu,bw\} \ldots\ldots (1)$$

You can use equation 2 to represent virtual machine instances in all private cloud environments, where r_i represents the virtual machine instance in the private cloud, and n represents the number of virtual machines owned:

$$PR=\{pr_1,pr_2,\cdots,pr_i,\cdots,pr_m\} \ldots\ldots (2)$$

Definition 2. resource model in the public cloud: a virtual machine instance can be described in the public cloud as equation 3, where price represents the cost of renting a public cloud instance:

$$pu=\{cu,bw,price\} \ldots\ldots (3)$$

use equation 4 to represent virtual machine instances in a common cloud environment, pu_j represents virtual machine instances in the public cloud, and k represents the number of public cloud instances that are leased:

$$PU=\{pu_1,pu_2,\cdots,pu_j,\cdots,pu_k\} \ldots\ldots (4)$$

Definition 3. resource model in a hybrid cloud: the resource model in a hybrid cloud consisting of a combination of public and private clouds can be defined as equation 5:

$$H=PR \cup PU \ldots\ldots (5)$$

Definition 4. a model of completion in a hybrid cloud.

the completion time of a workflow in a private cloud can consist of two parts, one of which is all tasks in a workflow i in the time to completion in a virtual machine, use equation 6 to represent the time it takes for a workflow task to execute in a virtual machine, using Ip to represent the completion time of a task in a virtual machine instance:

$$T_{comp}(t_i) = \frac{tl(t_i)}{cu(I_p)}, I_p \in PR \cup PU \ldots\ldots (6)$$

Another part of the time required for data transfer in a workflow task can be expressed using Equation 7, where represents the cost of data transfer between two workflow tasks, with representing a virtual machine instance as, means that two tasks are completed in the same virtual machine:

$$T_{comm}(t_i,t_j) = \begin{cases} \dfrac{tt(t_i,t_j)}{\min(bw(I_p), bw(I_q))}, & V_p \neq V_q \\ 0, & V_p = V_q \end{cases} \ldots\ldots (7)$$

Therefore, Equation 8 can be used to represent the task completion time in the hybrid cloud, where HT represents the total time it takes to perform a workflow task in the hybrid cloud, and PT refers to the time it takes to schedule tasks in the public cloud:

$$HT = \sum\nolimits_{i=1,j=i+1}^{n} \left(T_{comp}(t_i)\right) + T_{comm}(t_i,t_j)) + PT \ldots\ldots (8)$$

Definition 5. Spending model in a hybrid cloud: You can represent spending in a hybrid cloud with Equation 9, where represents the price spent on each shared cloud that you rent:

$$Cost = \sum_{i=1}^{m} P_v$$

Multi-Workflow Scheduling Strategy in a Hybrid Cloud

Multi-Workflow Scheduling Algorithm in a Private Cloud Environment

Because algorithms in private clouds do not need to consider price costs, algorithms in public clouds require cloud users to spend a lot of money to use. Therefore, to maximize the efficiency of using the private cloud, this section describes an Improved Ant Colony Optimization Workflow Scheduling (IACOWS) based on the goal of optimizing QoS [11]. Algorithm based on optimized ant colony algorithm, workflow scheduling optimization technology based on ant colony algorithm has been used for resource scheduling optimization problems, so the algorithm is used to carry out workflow scheduling in the private cloud environment, and the traditional ant colony algorithm will fall into the local optimal solution according to the optimization negotiation process to find the best match, so the algorithm introduces a random matching process in the process of performing workflow scheduling to prevent local optimal solution, the specific process of IACOWS is: 1) determine the task requirements and generate a random scheduling strategy; 2) Start the multi-ant mechanism based on resource availability, 3) start the negotiation process, 4) generate a feasible scheduling scheme, (5) evaluate the effect of the scheduling scheme through QoS, 6) generate a suitable scheduling scheme, and 7) generate a resource allocation and scheduling scheme.

The pseudocode for the iacows algorithm is as follows:

```
Algorithm 1. Improved Ant Colony Optimization Workflow Scheduling
1) Input: user task and resources
2) Output: appropriate task resource mapping
3) t1, t2, t3, tn user tasks
4) r1, r2, r3..rk cloud resource(VM)
5) ACO Parameters (α, β, ρ, m, max)
6) Fsrt: compute execution cost and time (feasible solutions)
7) Listrt: temporary storage list (contains feasible solutions)
8) Srt: minimum execution cost and time (best solution)
9) For
10) Input: Tasks t1, t2, t3, tn
11) check task validity
12) check resource availability r1,r2,r3..rn
13) Step 1
14) get resource availability
15) for (int i =0; i< n; i++)
16) if (rworkload = 0)
17) move into resource ready list
18) r = r[random() % k];
19) else
20) go to step1
21 end if
```

```
22) resource ready list
23) while users tasks not nill do
24) Multiple ant mechanisms started
25) compute task execution cost and time on available resources
26) ACO processes
27) Defined ACO Parameters (a, b, p, m, max)
28) While max criteria not meet do
29) Fsrt = for r1, rn/t1, tn
30) fs solution generation process start
31) fs contains estimated execution cost and time
32) test virtual matching up to max value(100)
33) Listrt = Timert (r1)
34) Srt = Sort (Listrt) //evaluation criteria
35) If (Stagnation) and (UsingSuspisionImprovement)
36) Use SuspisionCorrectionMethod
37) p (pheromone evaporation) //remove unnecessary solutions
38) select best solution (minimum execution cost and time)
39) mapping = Srt
40) resources dispatching //VMs to PMs
41) resource scheduling r1t2,r2t3,r3t1,rktn
42) execution process starts
43) task execution
44) task execution finish, release resources
45) finish
46) go to step 1
47) end
```

The goal of this algorithm is to find the optimal solution for efficient workflow scheduling. This method was inspired by the behaviour of ants in finding paths in the search for food. the virtual machine helps us generate random initial values, and finally finds the optimal scheduling strategy on the virtual machine through the optimization scheme.

The proposed QRAS code works as follows: The algorithmic process begins with the input of the user's task. System Analyzer then validates the task and lists it in a list that can be used. Start looking for idle resources and put them in the availability list. Due to the large number of available resources in the availability list, multiple colony systems initiate the negotiation process. It generates the initial solution by looking at all possible outcomes. The process continues to build the solution until the termination condition is met. These solutions are stored in a temporary list for evaluation. In addition, QoS-based assessments identify reasonable solutions, and others are sent to the evaluation list.

These standardized solutions are able to perform each task at an affordable cost and time. in addition, the placement of virtual machines has been formed based on the virtual mapping. it defines the optimal schedule by finding the best results in all solutions. the task execution process then begins, allocating the optimal workload to the resources. as the task execution ends, the resource is released to the available state. this process continues until the user completes the task.

Multi-Workflow Scheduling Algorithm in a Hybrid Cloud Environment

Since most of the enterprises currently use hybrid cloud forms to deploy task nodes, in order to save enterprise profits and improve the utilization of private cloud resources, it is necessary to maximize the use of private cloud resources in the enterprise, but also to ensure the efficiency of the execution time of workflow scheduling in the private cloud, so the ACAOWS algorithm using algorithm 1 prioritizes scheduling workflow tasks in the private cloud environment, when the execution time of workflow tasks exceeds the expected workflow scheduling time (WD, Workflow Deadline), Then use the public cloud environment to schedule part of the workflow. In this section, an Improved Hybrid Cloud Deadline-Constrained Cost Workflows Scheduling (IHCDCW) scheduling algorithm in a hybrid cloud environment is proposed, which solves the above problems very well.

The IHCDCW algorithm pseudo code is as follows:

Algorithm 2. Improved Hybrid Cloud Deadline-Constrained Cost Workflows Scheduling
1) Input: user task and resources
2) Output: appropriate task resource mapping
3) t1, t2, t3, tn user tasks
4) r1, r2, r3..rk cloud resource(VM)
5) ACO Parameters (α, β, ρ, m, max)
6) fsrt: compute execution cost and time (feasible solutions)
7) Listrt: temporary storage list (contains feasible solutions)
8) Srt: minimum execution cost and time (best solution)
9) For
10) If (jobs. size ! = 0)
11) Execute ACOWS // Execute Algorithm 1 to get the execution result in the private cloud
12) If (execute_time> wd)
13) For each subjob in jobi
14) Send subjob to public cloud // Schedule public cloud resources
15) End
16) If (Stagnation) and (UsingSuspisionImprovement)
17) Use SuspisionCorrectionMethod
18) End

Simulation Setup and Analysis

In this section, in order to verify the effectiveness, time cost, and expense cost of the IHCDCW algorithm in the hybrid cloud environment, the CloudSim [9] simulator is used to implement the HCDCW algorithm., which adds functions such as task clustering, resource configuration, and task scheduling for workflows. At the same time, it provides real cloud environment functions such as data centers, virtual machines, users, Cloudlets and bandwidth for configuration and customization. The CloudSim tool requires JDK version 1.6 or later, and CloudSim can be used using Eclipse or NetBeans.

This research work will build a hybrid cloud environment where a cloud service provider provides multiple charging mechanisms for virtual machine instances. Common charging mechanisms are on-demand charging, reservation charging and bidding charging, mainly considering the on-demand charging mechanism, that is, the virtual machine instance is charged according to a certain unit time (such as Google billing by the minute, AWS, Alibaba Cloud, etc. billed by the hour), and if it is not satisfied with the billing time, it is charged according to the whole calculation duration. This article uses the billing method of Alibaba Cloud in Table 1.

Table 1. Types and prices of virtual machine instances billed by Alibaba Cloud on-demand

instance type	vCPU	MEMORY (GB)	By Volume (hours)	Standard Catalog Monthly Price (USD)
universal ecs.g6.large	2	8	0.5	240
universal ecs.g6.xlarge	4	16	1.0	480
memory type ecs.r6.large	4	32	0.66	318
memory-based ecs.r6.xlarge	8	64	1.33	636
computational ecs.c6.large	2	4	0.39	187
computational ecs.c6.xlarge	4	8	0.78	374

The Montage dataset [10] is used in this lab, which is the workflow data suitable for the simulation environment, as shown in Figure 3, which is in the form of a Montage workflow, depending on the configuration of the simulation environment.

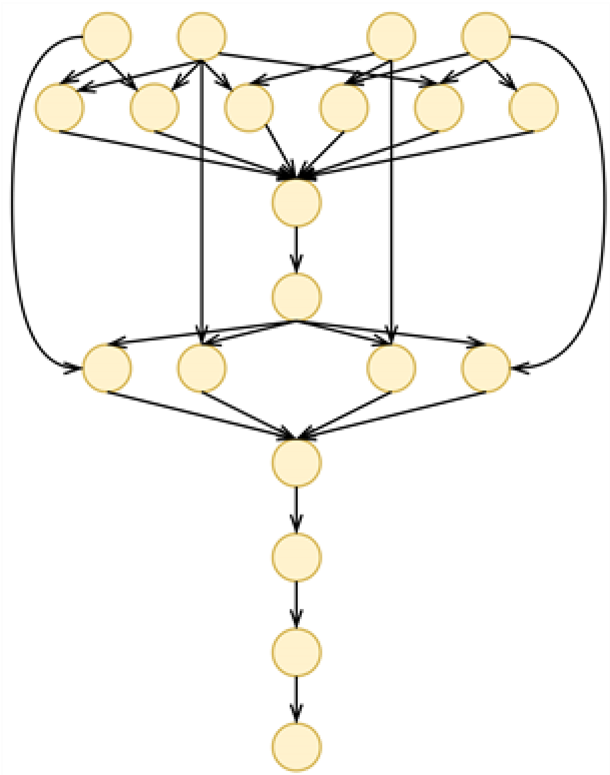

Figure 3. Montage dataset

The parameters for the virtual machine are set to table 2.

Table 2. Virtual machine parameter settings

Data Centre Setup	
Number of hosts	1
VMs Configuration	
Number of VMs	8
VM memory	1024
Bandwidth	4Mbps
Workload Environment	
Number of users	20
Numbers of cloudlets	1000
ACO Parameters Setup	
Number of iterations(max)	100
Number of ants(m)	10

Execution Time

This section compares the IHCDCW algorithm with the traditional MINMIN algorithm classical HEFT (Heterogeneous Earliest Finish Time). Figure 4 shows that when 100 workflow tasks are performed, the private cloud execution time increases with the increase of the wd maximum deadline. The execution time of the IHCDCW algorithm is significantly lower than that of the MINMIN and HEFT algorithm. Because IHCDCW makes full use of private cloud resources, it reduces the idle time of virtual machines in the private cloud and improves resource utilization.

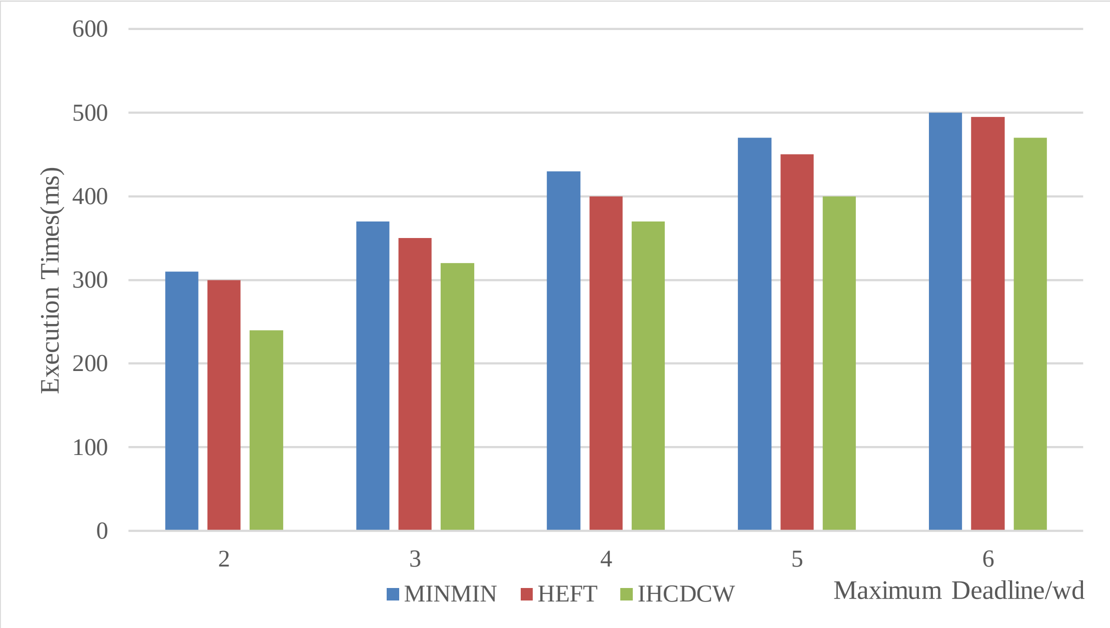

Figure 4. Execution time of 100 workflow tasks

Execution Costs

This section compares the IHCDCW algorithm with the classical HEFT (Heterogeneous Earliest Finish Time) [3] algorithm and the traditional MINMIN algorithm in terms of cost of expense. As shown in Figure 5, the IHCDCW algorithm saves a lot in execution costs compared to the classic HEFT algorithm.

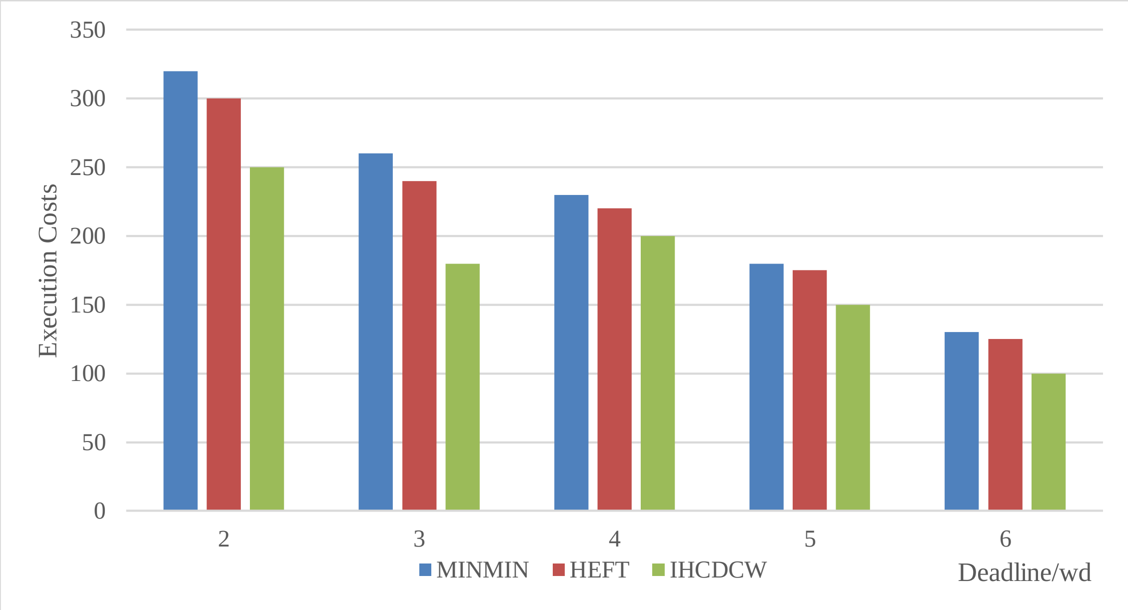

Figure 5. The execution cost of 100 workflow tasks

Conclusion

In this paper, the research goal is to optimize the increasing hybrid cloud workflow scheduling in enterprises, and proposes a dynamic multi-workflow optimization algorithm in the hybrid cloud environment. The algorithm IHCDCW proposed in this paper reduces the cost of expense and shortens the execution time of tasks for enterprises when using public cloud. However, due to the workflow tasks running in the private cloud in the enterprise, once scheduled to the public cloud, there will inevitably be problems such as security and privacy, and how to ensure the secure transmission from the private cloud to the public cloud can be considered in the future.

References

[1] Barbosa, F.P. and Charão, A.S. (2012) Impact of Pay-as-You-Go Cloud Platforms on Software Pricing and Development: A Review and Case Study. In: Murgante, B., et al., Eds., Computational Science and Its Applications—ICCSA 2012. Lecture Notes in Computer Science, 7336, 404-417. https://doi.org/10.1007/978-3-642-31128-4_30

[2] Tian Zhuojing, Huang Zhenchun, Zhang Yinong. A review on task scheduling methods in cloud computing environments[j]. computer engineering and applications, 2021, 57(2): 1-11.

[3] Topcuoglu, H., Hariri, S. and Wu, M.Y. (2002) Perfor-mance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing. IEEE Transactions on Parallel and Distributed Systems, 13, 260-274. https://doi.org/10.1109/71.993206

[4] Hwang, J.J., Chow, Y.C., Anger, F.D., et al. (1989) Scheduling Prece-dence Graphs in Systems with Interprocessor Communication Times. SIAM Journal on Computing, 18, 244-257. https://doi.org/10.1137/0218016
[5] Lin, C. and Lu, S.Y. (2011) Scheduling Scientific Workflows Elastically for Cloud Computing. 2011 IEEE 4th International Conference on Cloud Computing, 2011, 746-747. https://doi.org/10.1109/CLOUD.2011.110
[6] Zhou, J.L., Wang, T. and Cong, P.J. (2019) Cost and Makespan-Aware Workflow Scheduling in Hybrid Clouds. Journal of Systems Architecture, 100, Article No. 101631. https://doi.org/10.1016/j.sysarc.2019.08.004
[7] Ritchie, G. and Levine, J. (2003) A Fast, Effective Local Search for Scheduling Independent Jobs in Heterogeneous Computing Environments. Proceedings of the 22nd Workshop of the UK Planning and Scheduling Special Interest Group, Article ID: 15882331. https://www.semanticscholar.org/paper/A-fast%2C-effective-local-search-for-scheduling-jobs-Ritchie-Levine/cd4153cd3906897827cc2b537211c004ff95e89c
[8] Mo, L., Kritikakou, A. and Sentieys, O. (2019) Approximation-Aware Task Deployment on Asymmetric Multicore Processors. 2019 Design, Automation & Test in Europe Conference & Ex-hibition (DATE), 2019, 1492-1497. https://doi.org/10.23919/DATE.2019.8715077
https://doi.org/10.1109/eScience.2012.6404430
[9] Calheiros, R.N., Ranjan, R., Beloglazov, A., et al. (2011) CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Pro-visioning Algorithms. Software: Practice and Experience, 41, 23-50. https://doi.org/10.1002/spe.995
[10] Bharathi, S., Chervenak, A., Deelman, E., Mehta, G., Su, M. H., & Vahi, K. (2008, November). Characterization of scientific workflows. In 2008 third workshop on workflows in support of large-scale science (pp. 1-10). IEEE. https://doi.org/10.1109/WORKS.2008.4723958
[11] Tuba, M., & Jovanovic, R. (2013). Improved ACO algorithm with pheromone correction strategy for the traveling salesman problem. International Journal of Computers Communications & Control, 8(3), 477-485.