Blockchain-powered Service Migration for Uncertainty-aware Workflows in Edge Computing

Xiaolong Xu[1], Qingfan Geng[1], Hao Cao[1], Ruichao Mo[1], Shaohua Wan[2], and Lianyong Qi[3(✉)]and Hao Wang[4]

[1] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China
[2] School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan, China
[3] School of Information Science and Engineering, Qufu Normal University, China
[4] Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjϕvik, Norway

Abstract. In edge computing, the workflow is used to simulate and manage computing tasks as well as information exchange for compute-intensive and data-intensive application, which is convenient for the various complex process to work orderly. However, the resource conflict among cooperative works of multiple mobile edge computing (MEC) nodes by workflow, together with the service failure and the performance degradation, bring about additional uncertainties of scheduling strategies.Consequently, such uncertainties delay the completion of tasks and spoil the user experience. To deal with that issue, we propose a blockchain-powered resource provisioning (BPRP) method to design policies for workflows in the edge computing environment. Technically, we use the directed acyclic graph to indicate workflows of each edge node and regard its scheduling strategy as an individual gene to adapt to the following algorithm. Then, we use the non-dominated sorting genetic algorithm-III (NSGA-III) to optimize the workflow scheduling strategies on the basis of tasks' timely completion with good quality. A large number of experiments were carried out to verify the effectiveness of our method.

Keywords: Blockchain; Uncertainty-aware; Edge computing; Workflow; NSGA-III.

1 Introduction

In the cloud computing environment, many users share a resource pool and allocate resources through dynamic resource scheduling mechanism [1]. Therefore, compared with traditional data centers, the utilization rate of resources in cloud computing has been greatly improved. Nowadays, the cloud computing pattern has been effectively applied to solve complex scientific problems without human participation [2] [3]. This mode, as one of the construction technology of smart city, can improve work effectiveness and service quality to a great extent. However, cloud computing requires feedback from remote cloud data centers (CDC), which usually leads to long-distance round-trip delay, service interruption, and network congestion [4] [5]. With the dawn of the 5G era, the requirement of real-time in the intelligent network is increasing rapidly[6] [7]. Cloud computing seems unable to meet emerging performance requirements. In order to solve this problem, mobile edge computing is proposed as an extension of cloud computing to bring computing resource closer to the data source.

Workflow is a desirable application benefiting from cloud infrastructure [8] [9], which is a part of Computer Supported Cooperative Work (CSCW).A workflow application can automatically transfer documents, information and tasks among several participants following certain rules by the use of computers [10] [12]. However, in the procedure of workflow scheduling, there are some unstable factors, such as resource conflict, service failure and performance degradation, which will make the former static scheduling strategy no longer applicable, but requires a new real-time strategy. The unpredictable situations, uncertainties, are even more complicated in edge computing environment than in cloud computing environment due to the existence of control message among edge nodes [13] [14].

To solve the above problems, we consider introducing blockchain, a decentralized public ledger [15] [16], to record the status of edge nodes and the appropriate workflow scheduling strategies in real-time. We build a private blockchain which is not open to the public and only licensed nodes can participate in and view all data. The fundamental of a private blockchain is chain-data structure, and each data block records historical transaction or affair. Mobile devices can enhance their computing abilities by accessing edge serves (such as IoT Sensor Data Processing, which make edge computing become an effective solution for blockchain applications in mobile service. A private blockchain can also be synchronized globally, that means each edge node is able to know the current status of other nodes in real-time, thus making the scheduling strategies more intelligent and effective. At the same time, the consensus among each data block ensures the security and dependability of the blockchain, which brings technical guarantee for the sustainable development of the whole system.

This paper proposes a blockchain-powered resource provisioning method, named BPRP, which aims to minimize the processing time and energy consumption in the edge computing environment, and at the same time, optimize the workflow scheduling strategies by the use of blockchain. The contributions of this paper are as follows:

- In blockchain-based edge computing environment, the processing time model and energy consumption model.
- We propose a Blockchain-powered resource provisioning (BPRP) method to solve the uncertainty problem and optimize the workflow scheduling strategy.We also use NSGA-III to minimize the processing time and energy consumption.

– The results of the simulation show that our method effectively reduces energy consumption and execution time, and the uncertainty problem is solved commendably.

The rest of this paper is organized as follows. Section 2 gives basic concepts and definitions. Section 3 explains the method we use. Section 4 shows the performance of our method by experiments. Section 5 lists the relevant work. Section 6 summarizes the conclusions.

2 Model and Problem Formulation

2.1 Model

In this paper, our model is built on the basis on smart city and there are different base stations, which are defined as edge nodes, to provide computing resources for mobile devices. Suppose there are M mobile devices, denoted by $P = \{p_1, p_2, ..., p_M\}$, that generate workflows. Each workflow needs to schedule its tasks to different edge nodes to get adequate computing resources. A variable T_i is introduced to represent the task set of the $i-$th edge node, i.e., the workflow, and the workflow set of M mobile device is denoted as $T = \{T_1, T_2, ..., T_M\}$. To simulate the process of workflow scheduling, N edge nodes are used to provide computing resources and are denoted by $C = \{c_1, c_2, ..., c_N\}$.

In order to record the scheduling strategies of the workflows and ensure security, the blockchain is introduced and a blockchain-powered framework is proposed. Figure.1 shows the framework we proposed. The workflows generated by the mobile device are directly transmitted to different edge nodes for processing, and the edge nodes record the scheduling strategies. If some of the tasks in the workflows are unable to be processed by the edge nodes, they are transmitted to the data center.

Blockchain is a decentralized recording technology, and the scenarios of distributed ledger, which is based on blockchain are utilized in our framework. Compared to traditional accounting techniques, blockchain-based distributed ledger maintains an ever-increasing chain, which is only be added records and unable to tamper with existing records. Therefore, the security of data transmission can be guaranteed. In our framework, scheduling strategies and data information are considered to be scheduling transactions and all edge nodes support the architecture of blockchain. A block is created when a task is transmitted to a corresponding edge node. Besides, each block must also be verified for validity, and it can be quickly verified by calculating the hash value, which includes the information and the hash of the previous block. As shown in Figure.2(a), a block is generated based on a task, and then all edge nodes compete for recording this transaction. The winner act as a package node to package this transaction. Next, this edge node broadcast the transaction to other edge nodes to inform them that the transaction has been verified, and the transaction is appended to the blockchain in the form of the block after obtaining the consent of most edge nodes. The above process is shown in Figure.2(b) and Figure.2(c). As shown in

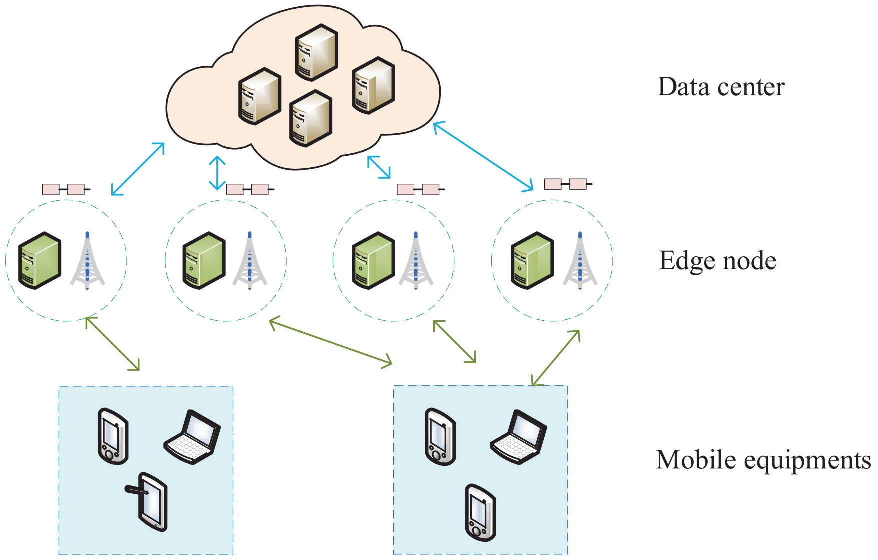
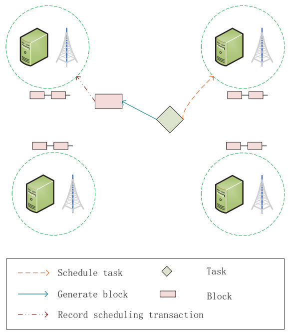


Fig. 1: A blockchain-based framework in edge computing.

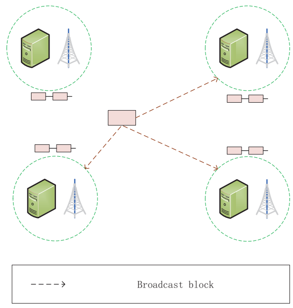
Figure.2(d), in the process of scheduling, due to the existence of uncertainty, which will be discussed in detail later, new blocks including remaining tasks are generated in the same way and join the blockchain.

In the edge network, the allocation of computing resources for each edge node is usually dynamic, which makes the original scheduling strategy less globally optimal after a period of time. This uncertainty is caused by service failure and performance degradation of the edge nodes. In this case, the affected tasks need to be rescheduled to obtain a new optimal scheduling strategy. Meanwhile, new blocks are generated in the same way and appended to the blockchain to update the scheduling strategies and the state of edge nodes. Service failure means that the edge nodes encounter some unpredictable failures, and the computing resources that were originally provided are no longer available. To ensure the completion time of the tasks, new scheduling strategies need to be developed to continue processing the affected tasks, including current and subsequent tasks. The edge nodes are usually able to continue to provide computing resources for the tasks when the performance of the edge nodes is degraded, but this situation delays the completion time of the tasks, resulting in worse user experience. Compared to the delayed time, the time to develop new strategies is ignored. Therefore, new strategies need to be developed to deal with performance degradation.

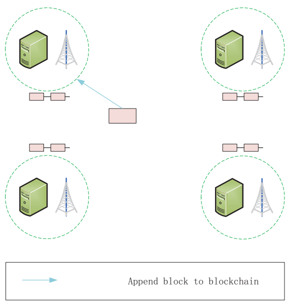
The blockchain is maintained by all edge nodes, and each node can be copied to obtain a copy of the complete or partial record, which is beneficial for the edge nodes to obtain each other's state information. Moreover, any changes in the blockchain are easily detected by each edge node, and it means that once the edge nodes generate uncertainty, other edge nodes can quickly detect and make new scheduling strategies.



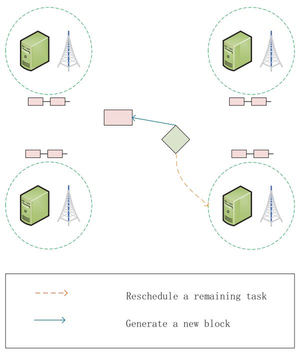
(a) compete for recording and generate block



(b) broadcast block for other miners



(c) append the block to the blockchain



(d) generate block when the uncertainty is generated

Fig. 2: Process of addressing the uncertainty in blockchain-based edge environment

2.2 Problem Formulation

In this section, the completion time and energy consumption of workflow scheduling are considered. Besides, when the edge nodes generate uncertainty, the success rate of processing the affected tasks is quantified.

For each workflow T, a variable E is introduced to define the relationship between the tasks in the workflow, and $E = (t_i, t_j)$ represent that the task t_i need to be completed before the task t_j. In order to guarantee the quality of service(QoS) of users, low-latency services need to be provided and each workflow T is given a deadline D. A triple-tuple $h_i = (T_i, E_i, D_i)$ is also introduced to represent the relationship of $i-$th workflow T_i and its deadline.

First, a workflow is analyzed in this section, and the start time and completion time of each task in the workflow are discussed. For the first task, its earliest start time is set as follows:

$$FT(t_1) = 0. \tag{1}$$

Based on the parent tasks of each task, the earliest time of each subsequent task in the workflow is calculated in turn. The parent tasks of $m-$ th task are defined below:

$$pa(t_n) = \{t_m | t_m \in E(t_m, t_n)\}. \tag{2}$$

Then, the earliest start time of the task t_m is calculated by:

$$FT(t_m) = \max(FT(t_n) + TT(t_n) + ET(t_n)), t_n \in pa(t_m). \tag{3}$$

where $TT(t_n)$ denotes the transmission delay between the parent tasks and the task t_m, $ET(t_n)$ denotes the execution delay of the parent tasks, and the earliest start time of the task t_m is affected by these two variables. According to the execution time of the t_m, its earliest completion time is also calculated by:

$$FFT(t_m) = FT(t_m) + ET(t_m). \tag{4}$$

In the process of workflow scheduling, the last task t_{last} of the workflow has the latest completion time $LT(t_{last})$, which is determined by the deadline of the workflow.

$$LT(t_{\text{last}}) = D. \tag{5}$$

Before calculating the latest deadline for each task, the subtask set of its is first defined.

$$ch(t_m) = \{t_m | t_m \in E(t_m, t_n)\}. \tag{6}$$

Then, the latest completion time of the task t_m is calculated by:

$$LT(t_m) = \min(LT(t_n) - TT(t_n) - ET(t_n)), t_n \in ch(t_m). \tag{7}$$

where $TT(t_n)$ denotes the transmission time of the task t_m and its subtasks, $ET(t_n)$ denotes the execution time of the subtasks.

However, during the actual process of workflow scheduling, the actual completion time $RC(t_m)$ of the task t_m is known. Each task in the workflow also has its own deadline, which can be found by the critical path algorithm. Assume that the task sequence to be processed on the critical path q is $q = \{q_1, q_2, ..., q_x\}$, and the task q_y corresponds to the task t_m which is discussed above. Therefore, the latest deadline for t_m is calculated by:

$$D(q_y) = \frac{FFT(q_y) - FT(q_1)}{FFT(q_x) - FT(q_1)} \times (LT(q_x) - FT(q_1)). \tag{8}$$

$$D(t_m) = D(q_y). \tag{9}$$

The actual completion time of each task needs to be constrained to be less than its deadline. Besides, the energy consumption of workflow scheduling also needs to be considered in our model, and a variable $f_{m,k}$ is introduced to determine the scheduling strategy.

$$f_{m,k} = \begin{cases} 1, \text{if } t_m \text{is transmitted to } c_k, \\ 0, \text{otherwise.} \end{cases} \tag{10}$$

If the task t_m is transmitted to th e edge node c_k to obtain sufficient resources, $f_{m,k} = 1$, otherwise $f_{m,k} = 0$. Thus, the energy consumption of workflow scheduling is calculated.

$$EN = \sum_{m=1}^{M} \sum_{k=1}^{N} f_{m,k} \cdot tc_{m,k} + \sum_{m=1}^{M} ec(t_m). \tag{11}$$

where $tc_{m,k}$ denotes the transmission energy consumption between the task t_m and the edge node c_k, $ec(t_m)$ denotes the execution energy consumption of the task t_m and it is determined by the size of the data.

Suppose that the set of all tasks affected by the uncertainty is $F = \{t_1, t_2, ..., t_U\}$, and there are U tasks. After rescheduling the affected tasks, a variable w_v is introduced to determine whether the affected tasks can be successfully resolved or not.

$$w_v = \begin{cases} 1, RC(t_v) <= D(t_v), \\ 0, \text{otherwise.} \end{cases} \tag{12}$$

If the actual completion time of the task t_v is less than its deadline, it can be successfully solved, $w_v = 1$, otherwise $w_v = 0$. Therefore, the success rate of processing uncertainty is calculated.

$$R = \frac{1}{U} \sum_{v=1}^{U} w_v. \tag{13}$$

Our goal is to minimize the completion time and energy consumption and maximize the success rate simultaneously. This multi-objective optimization problem is defined.

$$\min(RC(t_{last})), \min(EN), \min(\frac{1}{R}). \tag{14}$$

$$s.t. \forall t_m, RC(t_m) \leq D(t_m). \tag{15}$$

3 A Blockchain-powered Resource Provisioning Method

In our framework, a blockchain-powered resource provisioning (BPRP) is proposed to solve the uncertainty in the workflow scheduling process. This method needs to solve a multi-objective optimization problem and the non-dominated genetic algorithm III (NSGA-III) perform well in this problem.

3.1 Initialization

Each task in the workflow has a strategy to decide which edge node it achieves computing resources. Suppose there are M tasks, denoted as $T = \{t_1, t_2, ..., t_M\}$, and their strategies are mapped into the decision space. In NSGA-III, the strategy of each task represents a gene, and the strategy set of a workflow, called by chromosome, consists of all the tasks. The size of the parent population in NSGA-III is determined by the decision variables, and the solutions to this multi-objective problem are found after a certain number of iterations. The fitness functions of the three metrics are given in Eq.(13), and the constraints are given in Eq.(14).

3.2 Optimal strategy selection using NSGA-III

First, the number of individuals in the parent population A is set to M, which is illustrated in the previous subsection. After using the genetic operator, which includes the selection operator, the recombination operator and the mutation operator, an offspring population B is obtained from the parent population A. Then, the population A and the population B are mixed to obtain a population C of size $2M$. The population is non-dominated sorted into layers of individuals with Y layers. The individuals of each layer are sequentially appended to a new population D until the size of D is greater than M, and now the layer is marked as K. Some individuals are selected from the $K-$th layer such that the size of the population D is equal to M.

The ideal point set is important in NSGA-III, and it is derived by calculating the minimum of each fitness function. The scalar formula for each fitness function is given:

$$RC(t_{last})^m = RC(t_{last}) - RC_{\min}(t_{last}). \tag{16}$$

$$EN^m = EN - EN_{\min}. \tag{17}$$

$$\frac{1}{R}^m = \frac{1}{R} - \left(\frac{1}{R}\right)_{\min}. \tag{18}$$

where $RC(t_{last})^m$, EN^m, $\frac{1}{R}^m$ denote the three fitness functions respectively. Then, the extreme points of the fitness functions are found by the function ASF. The specific formulas are below:

$$RC(t_{last})^n = \frac{RC(t_{last})^m}{\lambda_{rc}}. \tag{19}$$

$$EN^n = \frac{EN^m}{\lambda_{en}}. \tag{20}$$

$$\frac{1}{R}^n = \frac{1}{\lambda_r} \cdot \frac{1}{R}^m. \tag{21}$$

where λ_{rc}, λ_{en}, λ_r are the intercepts on the corresponding axes.

Since NSGA-III is a multi-objective evolutionary algorithm based on reference points, individuals must be associated with reference points and it is able to completed through recursion. After the corresponding divided reference points are determined, the reference point vector, that is, the line from the reference point to the origin needs to be constructed. Then, the closest reference points to each population are found and the shortest distance is recorded.

Finally, the reference points are filtered. Suppose the number of individuals exceeds M for the first time at the $FK-$th layer. The reference points at the $FK-$th layer that is least referenced by other lower layers are found recursively, and the reference number of $j-$th point is recorded as p_j. If $p_j > 0$, the closest reference point to this is selected. If $p_j = 0$, and there is no individual at the $FK-$th layer is referenced to the reference point, the reference point is deleted. Otherwise, the individual closest to the reference point is selected and appended to the next generation population.

4 Experimental Results and Analysis

4.1 Experiment setup

In our experiments, the edge nodes are supposed to consist of a virtual machine(VM) group, and the VMs have the open power, idle power, and operating power. Each edge node is assumed to provide limited computing resources, and once the resource is occupied, the tasks need to be transferred to other edge nodes to obtain adequate computing resource. The workflows with a different number of tasks are also selected in our experiments, and the number of tasks is 10, 20, 30, 40, 50 and 60. Besides, the bandwidth between the mobile devices and the edge nodes is also determined. The specific parameters are illustrated in Table 1.

To simulate the uncertainty in workflow scheduling, the tasks affected by uncertainty are set in the workflow. When uncertainty occurs, the affected tasks need to be rescheduled to generate new strategies. The deadline is usually determined by the theoretical completion time of the workflow, and it is set to κ times the theoretical time. Based on the actual completion time of the affected tasks

Table 1: Parameter settings

Parameters	Value
The power of open VM α	0.25 KW
The idle power of VM β	0.02 KW
The working power rate of VM χ	0.05 KW
The processing power of edge node ω	2000MHz
The WAN bandwidth VM η	1000mbps

and the theoretical deadline of them, the success rate of solving the uncertainty is experimentally simulated.

To demonstrate the superiority of our method in dealing with uncertainty, three other methods are selected in our experiments. The first method is Benchmark and an optimal scheduling strategy is generated at first. However, new scheduling strategies are not developed when the uncertainty is generated in the edge nodes. The second method is First Fit(FF), which in turn looks for the edge nodes that are closer to get adequate computing resources and it reschedules the workflow when the uncertainty is generated. Similar to the second method, Next Fit(NF) request computing resources from the edge node which provides resource last time, instead of requesting the closer edge nodes.

4.2 Performance evaluation

In this subsection, the performance of BPRP, Benchmark, FF, NF is evaluated, and the three metrics are the success rate, the completion time and the energy consumption.

Comparison of the Success Rate Figure 3 shows the success rate of four different methods in solving uncertainty. Compared with the other three methods, BPRP has a higher success rate in a shorter time, and it improves the quality-of-service(QoS). As time increases, the success rate of each method eventually reaches 1. However, it means a higher latency service and it makes no sense in real life.

Comparison of the Energy Consumption Figure 4 shows the energy consumption of these methods when processing workflow with a different number of tasks. It is obvious that our method has a lower energy consumption. When the uncertainty is generated, our method immediately senses and reschedules the workflow to avoid meaningless waste of resources.

Comparison of the Complete Time The completion time of these four methods is illustrated in Figure 5. By comparison, it is observed that our method

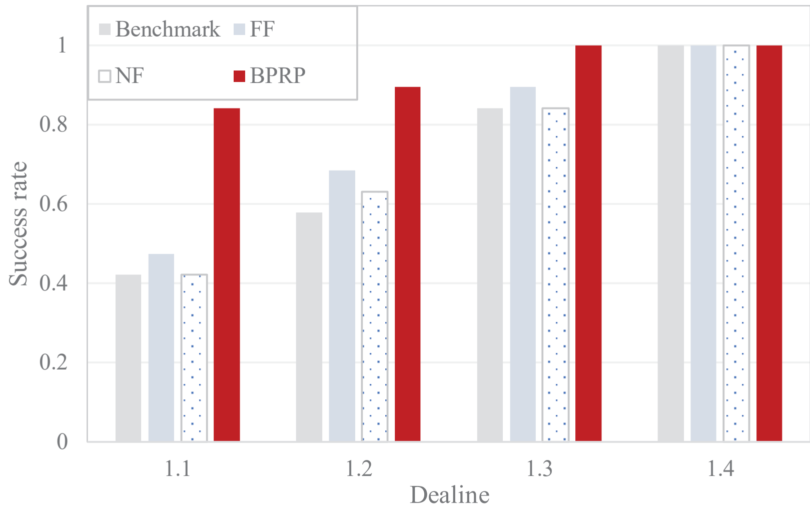


Fig. 3: Comparison of success rate.

is able to process the workflow in a shorter time. Due to the introduction of blockchain, other edge nodes quickly perceive and develop a new optimal strategy when the uncertainty is generated in an edge node. It greatly reduces the downtime, and then the uncertainty is more likely to be resolved.

5 Related work

In the cloud computing environment, workflow scheduling is faced with the unavoidable uncertainties, while effective scheduling method can provide a higher quality of service (QoS) for cloud users. WenAn Tan et al. proposed a workflow scheduling algorithm based on trust service, which uses the combination of direct trust and recommendation trust to measure trust [7].

Huangke Chen et al. proposed a real-time workflow scheduling combining active and passive scheduling, their method is named as PRS, it minimizes the uncertainty in the real-time workflow scheduling [2]. This scheduling strategy's particularity lies in that most unfinished tasks wait in task pool, instead of waiting for virtual machine (VM), and only tasks which have already mapped to VM are allowed to wait for VM. In [7], they also proposed an uncertain online scheduling algorithm with time constraints. It reduces the spread of uncertainty by controlling and adjusting the waiting tasks on the server and effectively improves the resource utilization in cloud computing.

Considering the great applicability of blockchain in decentralization and reliability, [8] proposes a method of sharing the origin of scientific workflow in blockchains. The sharing of the origin of scientific workflow enhances the cooperation of distributed research.

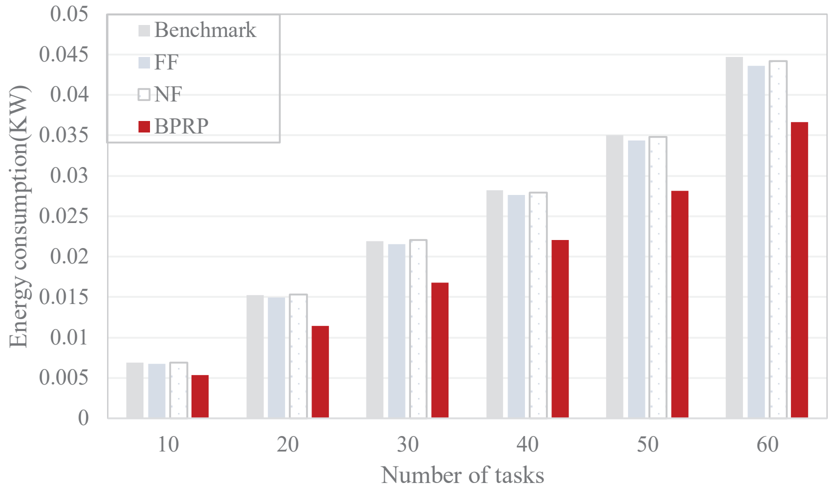


Fig. 4: Comparison of energy consumption.

It is mentioned that due to the performance changes of virtual machines, many scheduling interruptions may occur and pre-computed baseline scheduling may not be performed [2]. To this end, the authors of [9] propose a framework for virtual machine management coordination. Through the migration of the virtual machine, the network communication cost can be reduced to the greatest extent.

6 Conclusion

Nowadays, cloud computing is widely used by virtue of its automated network management function. However, the information feedback of remote cloud data center (CDC) can not meet the requirements of 5G applications with low latency and high bandwidth. In order to solve this problem, mobile edge computing is proposed as an extension. In addition, there are uncertainties such as resource conflict, service failure and performance degradation in cloud computing workflow scheduling process. To solve these uncertainties, we propose a blockchain-powered resource provisioning (BPRP) method to make policies for workflows in the edge computing environment. Non-dominated sorting genetic algorithm III (NSGA-III) is used to optimize the workflow scheduling strategy to ensure that tasks are completed in a timely and high-quality manner. The experimental results show that BPRP can successfully reduce the uncertainty of workflow scheduling and minimize processing time and energy consumption. This method can determine the capacity of each edge node according to the actual situation in the application.

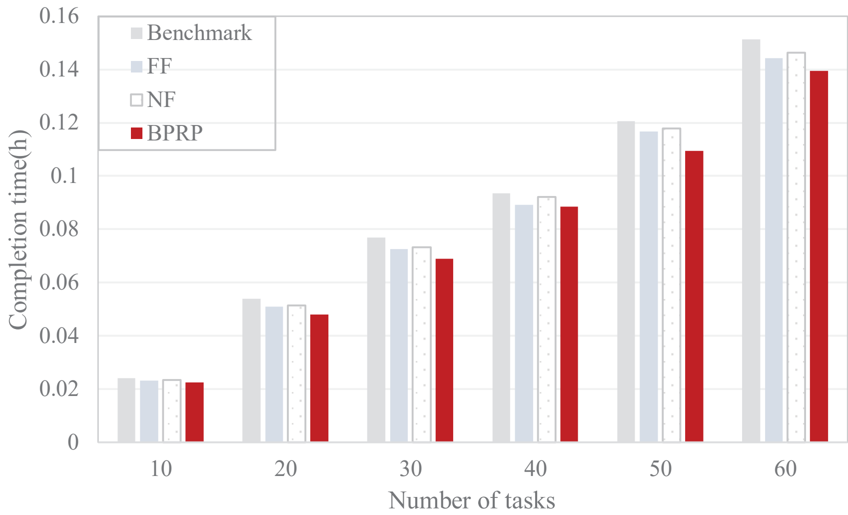


Fig. 5: Comparison of the completion time.

Acknowledgment

This research is supported by the National Science Foundation of China under grant no. 61702277.

References

1. Xu Chen, Wenzhong Li, Sanglu Lu, Zhi Zhou and Xiaoming Fu, Efficient Resource Allocation for On-Demand Mobile- Edge Cloud Computing, IEEE Transactions on Vehicular Technology, vol.67, no.9, pp.8769-8780, 2018.
2. Huangke Chen, Xiaomin Zhu, Guipeng Liu and Witold Pedrycz, Uncertainty-Aware Online Scheduling for Real-Time Workflow in Cloud Service Environment, IEEE Transactions on Services Computing, pp.1-1, 2018.
3. Shunmei Meng, Song Wang, Taotao Wu, Duanchao Li, Taigui Huang, Xiaotong Wu, Xiaolong Xu and Wanchun Dou, An Uncertainty-aware Evolutionary Scheduling Method for Cloud Service Provisioning, 2016 IEEE International Con- ference on Web Service (ICWS), pp.506-513, 2016.
4. Dinan Fakhri, Kusprasapta Mutijarsa,Secure IoT Communication using Blockchain Technology, 2018 International Symposium on Electronics and Smart Devices (ISESD), pp.1-6, 2018.
5. Madhusudan Singh ; Abhiraj Singh ; Shiho Kim, Blockchain: A game changer for securing IoT data, 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), pp.51-55, 2018.
6. Zehui Xiong, Yang Zhang, Dusit Niyato, Ping Wang, Zhu Han, When Mobile Blockchain Meets Edge Computing, IEEE Communications Magazine, vol.56 , no.8, pp.33-39, 2018.

7. An Tan, Yong Sun, Ling Xia Li, GuangZhen Lu, Tong Wang, A Trust Service-Oriented Scheduling Model for Workflow Applications in Cloud Computing, IEEE Systems Journal, vol.8 , no.3, pp.868-878, 2014.
8. Wanghu Chen, Xiaoyan Liang, Jing Li, Hongwu Qin, Yuxiang Mu, Jianwu Wang, Blockchain Based Provenance Sharing of Scientific Workflows, 2018 IEEE International Conference on Big Data (Big Data), pp.3814-3820, 2018.
9. Richard Cziva, Simon Jouet, David Stapleton, Fung Po Tso and Dimitrios P. Pezaros, SDN-based Virtual Machine Management for Cloud Data Centers, IEEE Transactions on Network and Service Management, vol.13, no.2, pp.212-225, 2016.
10. Yang J, Xu X, Tang W, et al. A Task Scheduling Method for Energy-Performance Trade-off in Clouds. //2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, pp.1029-1036, 2016.
11. Xu X, Dou W, Zhang X, et al. EnReal: An energy-aware resource allocation method for scientific workflow executions in cloud environment. IEEE Transactions on Cloud Computing,vol.4, no.2, pp.166-179, 2015.
12. Qi L, Meng S, Zhang X, et al. An exception handling approach for privacy-preserving service recommendation failure in a cloud environment. Sensors, vol.18, no.7, pp.2037, 2018.
13. Wen Y, Liu J, Dou W, et al. Scheduling workflows with privacy protection constraints for big data applications on cloud. Future Generation Computer Systems, 2018.
14. Qi L, He Q, Chen F, et al. Finding All You Need: Web APIs Recommendation in Web of Things Through Keywords Search. IEEE Transactions on Computational Social Systems, 2019.
15. Xu X, Liu Q, Zhang X, et al. A Blockchain-Powered Crowdsourcing Method With Privacy Preservation in Mobile Environment. IEEE Transactions on Computational Social Systems, 2019.
16. Lee H., Ma M. (2019) Blockchain-Based Mobility Management for LTE and Beyond. In: Wang G., Feng J., Bhuiyan M., Lu R. (eds) Security, Privacy, and Anonymity in Computation, Communication, and Storage. SpaCCS 2019. Lecture Notes in Computer Science, vol 11611. Springer, Cham.