Computer Communications 150 (2020) 413–420

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Migration strategy of cloud collaborative computing for delay-sensitive industrial IoT applications in the context of intelligent manufacturing

Ke Wang

Department of Information Engineering, Zhengzhou Tourism College, Zhengzhou City, Hennan Province 451464, China

A R T I C L E I N F O

Keywords:
Intelligent manufacturing
Industrial Internet of Things
Cloud cooperation
Computing migration

A B S T R A C T

In the context of intelligent manufacturing, machinery and equipment in the industrial manufacturing process form the "industrial Internet of Things." In this process of interlocking production, the requirements for sensor data delay typically reach the millisecond level. Once the data is delayed, the equipment will be shut down, which will make the production difficult or dangerous. In the context of intelligent manufacturing, local computers have been unable to complete calculations and decisions quickly and on time for the huge computing demands. Therefore, the cloud computing migration mode needs to be introduced, but cloud computing migration will cause additional delays. Based on the above problems, this paper designs a cloud cooperative migration strategy based on the information exchange structure of the industrial Internet of Things and the delay mechanism caused by the migration. The feasibility of selecting the optimal migration strategy based on task partitioning is verified by simulation.

1. Introduction

In the history of mankind, it has experienced three industrial revolutions. Currently, it has entered the fourth industrial revolution (Industry 4.0). Its main feature is to make full use of Internet technology, database technology, embedded technology, wireless sensor network, machine learning and other multi-domain technologies to achieve the intelligent, remote measurement and control transformation of manufacturing. It is also known as the modern information technology era. Its core is "Internet of Things + Intelligent Manufacturing" [1]. The development of the new generation of information technology in the new century has promoted the process of informationization in the manufacturing industry, and triggered an industrial revolution centered on smart manufacturing on a global scale. Intelligent manufacturing workshops put forward higher requirements for the agile response speed of the production process and the high efficiency of management decisions.

In the industrial wireless network system, various micro wireless sensor nodes are installed on the machine. Each sensor node typically contains sensors, wireless communication devices, micro-computing units, and power supplies. Sensors are used to collect various data such as temperature, humidity, pressure, and illumination. The wireless communication device is configured to send or receive a message, and the computing unit performs simple processing on the data collected or received by the computing unit. Finally, a sink node collects and analyzes the data from each sensor node. For a large industrial manufacturing process, it contains thousands of sensors. The amount of data summarized in the end is huge, and the amount of data processing and calculation brought by it is also a geometric growth of the process. Traditional servers and workstations cannot meet the above calculation requirements [2]. Related scholars proposed a low complexity task migration algorithm based on Lyapunov optimization theory, while reducing the energy consumption and task execution time of intelligent mobile terminals [3–5]. Compute tasks are migrated to various types of resource-rich cloud processing tasks, thereby reducing the resource consumption that is required for the calculation itself and saving energy [6,7]. The computational migration technique migrates computational tasks to large nodes with strong computing power on the transmission path for processing. However, migrating tasks to cloud virtual hosts for processing will introduce large additional delays (network latency and virtualization latency), making it difficult to meet the needs of delay-sensitive industrial manufacturing. This is reflected in the contradiction between real-time communication and business delay [8]. Therefore, in the actual situation, it is necessary to meet the computing needs of the industrial Internet of Things, and to minimize the communication delay, which is the core difficulty in calculating the choice of migration strategy.

This paper first analyzes the delay factors caused by computational migration for the industrial characteristics of the Internet of Things and the network information exchange structure in the context of intelligent manufacturing. Subsequently, a cloud-based computing migration strategy is designed. Finally, the different task division modes are compared, and the feasibility of optimal strategy selection is verified by simulation.

E-mail address: wangke_2107@163.com.

https://doi.org/10.1016/j.comcom.2019.12.014
Received 15 October 2019; Received in revised form 30 November 2019; Accepted 5 December 2019
Available online 9 December 2019

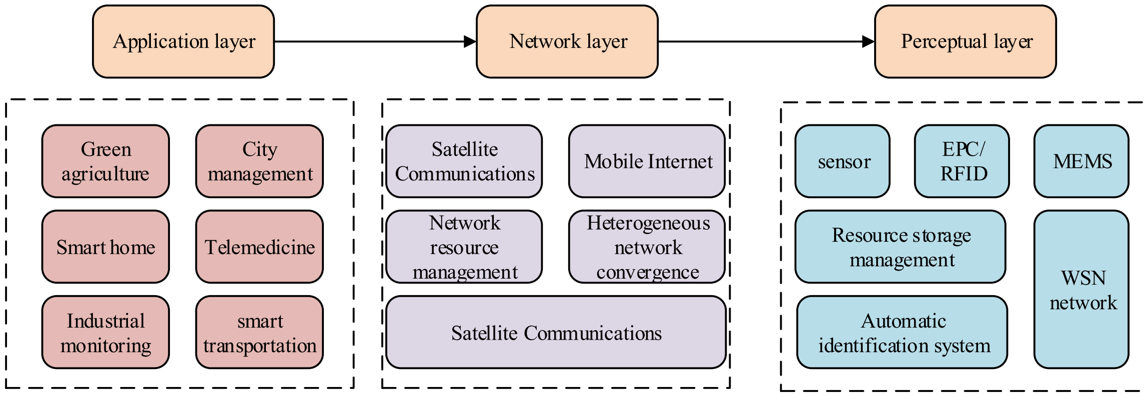

Fig. 1. Three-tier architecture of traditional Internet of Things.

2. Industrial IoT architecture under the background of intelligent manufacturing

2.1. Traditional internet of things structure

The Internet of Things (IoT) refers to a network that uses various information sensing devices to connect with the Internet to realize a comprehensive network of people, people and things, and things without time and geographical restrictions [9]. This can be divided into three levels according to the bottom-up process: the perception layer, the network layer and the application layer, as shown in Fig. 1.

The perception layer is at the bottom of the entire architecture and is the core to realize the comprehensive perception of the Internet of Things. It consists of various sensing devices, such as RFID (radio frequency identification) electronic tags and readers, QR code tags and recognizers, various types of sensors, cameras, GPS, and so on. Its main role is to identify connected objects, collect and capture various monitoring information in real time. If the Internet of Things is likened to a person, then the level of the perception layer is equivalent to the skin and facial features of the human body [10].

The network layer is located in the middle of the entire architecture and is responsible for deep integration of various communication networks (telecom network, mobile communication network, intranet, satellite network, wide-area network, small local area network, etc.) with the Internet, and is used to acquire the sensing layer. The information is passed to each network, usually over long distances. In addition, under the support of large data centers, cloud computing centers and other platforms, it is also responsible for the intelligent processing of the collected massive information; the status is equivalent to the human nerve center and brain. The application layer belongs to the uppermost layer of the entire architecture, and builds various intelligent application platforms for various practical business needs. So, different users can use the analyzed and processed information to provide a rich and intelligent solution [11], such as smart home, smart medical, intelligent building, intelligent transportation, smart agriculture, smart city, remote meter reading, etc.; the status is equal to the social division of labor.

2.2. Industrial IoT architecture in the context of intelligent manufacturing

Under the background of the birth of IoT technology and the rapid development of the Internet, the traditional industrial field has also ushered in deep reforms under the impetus of Internet of Things technology. DCS refers to a modern industrial control system that monitors and controls industrial and agricultural production processes based on fieldbus technology. It integrates computer technology, communication technology and control technology. At present, DCS is widely used in many fields such as modern science and technology, industrial production, national defense, meteorology, etc. It is one of the most mature and most widely used measurement and control system architectures [12]. A typical DCS architecture is shown in Fig. 2. The architecture takes the microcontroller as the core, and uses the fieldbus technology to connect the measurement and control terminals to each other, and connects to the Internet through the main control station module, so that it has the functions of monitoring and control.

The DCS is divided into three layers from the architecture: the measurement and control terminal layer, the network communication layer, and the business logic layer. The measurement and control terminal layer is composed of each terminal equipment node distributed in the measurement and control site, and completes the real-time collection of field data and equipment control. This is responsible for returning data to the upper layer through fieldbus technology and executing commands from the upper layer. The network communication layer is responsible for the mutual communication between the measurement and control terminal layer and the business logic layer, and is responsible for the conversion between the underlying field bus protocol and the high-level network communication protocol. The business logic layer is at the top of the overall architecture. It usually consists of an operator terminal, an engineer terminal, and an application server. The operator terminal is responsible for monitoring and controlling the actual measurement and control site [13]. The engineer terminal is responsible for managing the system hardware and software resources and other important components. The application server is responsible for providing network communication services and data storage functions for the entire architecture.

2.3. Industrial IoT information exchange structure

In recent years, the industrial modernization process has intensified; traditional industrial control has been improved and developed rapidly; and industrial automation and production line intelligent unmanned has become a new development trend. As production equipment and process requirements become higher and higher, which makes production equipment and production lines more and more complex, the analysis on remote intelligent control and regulatory is becoming more and more important. The Internet of Things remote data transmission system can automatically upload device operation data to the Internet. It is aggregated, processed and stored in the database in the Internet Data Service Center [14]. Then, it helps to transform and disassemble the data information. Finally, it is displayed in the form of intuitive graphs, line graphs, pie charts and tables, so that engineering monitors can quickly obtain data transmission information and accurately control the production and operation of machinery and equipment.

With the continuous evolution and transformation of intelligent manufacturing control technology, various types of production equipment have become more advanced and intelligent. The demand for remote data transmission technology has also become more and more urgent, which will bring more opportunities for the industrial IoT data

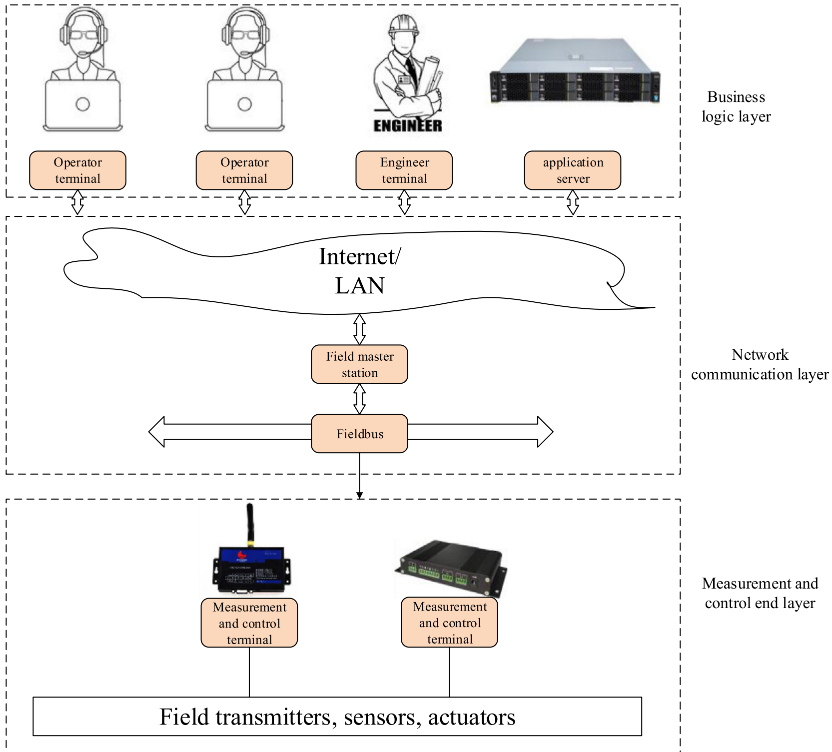

Fig. 2. The architecture of DCS.

transmission field. With the advent of more modern technologies such as cloud platform, intelligent control, and intelligent management, the data transmission method has gradually evolved into an effective monitoring method and a new type of resource management. In the future, IoT remote data transmission technology will evolve towards integration, intelligence and production. In an enterprise information network consisting of information management layer, process monitoring layer, and field device layer (as shown in Fig. 3). Ethernet is already the de facto standard protocol in the information management layer, and the process monitoring layer network also basically transitions from serial communication to Ethernet technology [15]. Therefore, industrial users hope that Ethernet technology will directly extend from the upper information management level to the process monitoring level and field device level of the vertical communication, thereby providing a network foundation for information exchange between different layers of the automation system.

3. Industrial IoT delay analysis

3.1. Industrial IoT performance requirements for data latency

Data latency, which refers to the time difference between the data being generated from the data source and being received by the data target, is also one of the basic indicators for evaluating network performance. With the improvement of the precision and speed of modern industrial production, the requirements for data delay in industrial wireless networks are gradually increasing. On the one hand, the shorter delays help the system react more quickly and accurately, thereby improving product quality and productivity; On the other hand, in industrial systems, the collected data is often only valid for a short period of time, and the expired data is not only useless or even harmful. The need for data latency can often be divided into two broad categories: soft real-time systems and hard real-time systems. In a soft real-time system, real-time performance guarantees a predetermined probability, that is, a small amount of over-time delay is allowed. In a hard real-time system, the end-to-end delay boundary must be strictly determined, and the delay of information beyond the boundary means the failure of the system. Industrial communication systems must meet the real-time requirements of the technical process. Depending on the application of the system, real-time requirements can be divided into three different orders of magnitude. For example, in process automation, data acquisition requires an appropriate time limit. The typical response time is about 100 ms. When the response time changes, its performance does not deteriorate significantly. In motion control systems, high determinism and low jitter of data transmission delay must be guaranteed, with a response time of less than 1 ms and jitter of less than 1 μs, as shown in Table 1.

Table 1
Demand for industrial communication delay.

Traffic type	Response time	Shaking
Soft Rt	10–100 ms	≥100%
RT	1–10 ms	≥15%
IRT	<1 ms	<1 μs

At the field device level, two major issues must be considered, which are predictability and real-time. In particular, predictability refers to reading and writing variables in a certain time interval; real-time refers to the maximum time allowed for an event communication task to run. Periodic data exchange between field-level field devices. The data length is small; the update period is short; the jitter is small; and the enterprise management layer and the control monitoring layer penetrate into the field device level, which will bring huge data transmission amount. The quality of service of the system must be guaranteed, and the significant yardstick for judging the quality of its service is the end-to-end delay.

3.2. Calculating network delay caused by migration

The modern industrial Internet of Things cloud-based computing migration solution will bring additional communication overhead to mobile terminals. For example, in industrial communication, it is firstly

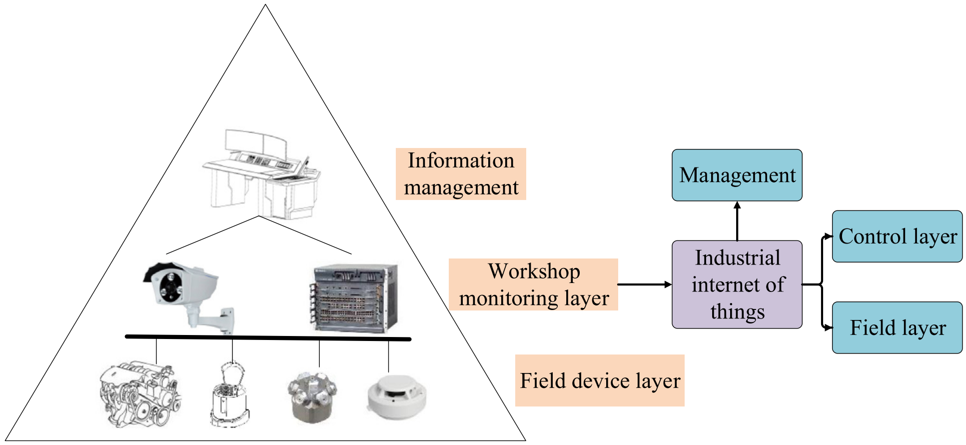

Fig. 3. Industrial IoT information exchange structure.

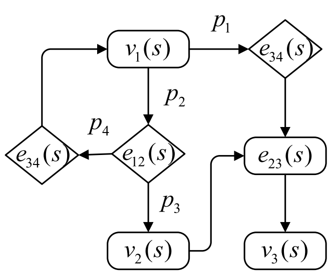

Fig. 4. Network delay system model.

necessary to extract the operation information of the technician and then transmit it to the corresponding control network for processing. Of course, before submitting to the cloud, the user operation information will be extracted, encoded and compressed firstly, and then uploaded. After the control network processes the user operation information, the corresponding feedback information is fed back to the client in the form of a stream. The entire process from collection, uploading, processing, and returning information to the client must be completed in milliseconds. Only in this way can the normal interaction time between the user and the device be met. Therefore, even if the computing resources in the cloud are sufficient, the network delay introduced by the computing migration in the cloud environment is still a key factor in determining whether industrial production control is smooth.

In large-scale network systems, the cumulative distribution function (CDF) of information delay is a very important indicator to measure the internal reliability of the system. Also in industrial communication networks, if control information or feedback information is not successfully delivered due to delays, network performance may be degraded or even destructive [16,17]. Therefore, when designing a reliable communication protocol, a powerful tool is needed to accurately model and analyze the end-to-end delay profile. In the design and optimization of large-scale industrial networks, it is necessary to fully capture the end-to-end delay characteristics and distribution according to the statistical characteristics of network components and protocols. This establishes an accurate end-to-end delay model to provide network reliability and quality of service assurance.

A typical delay system can be represented by a series of component nodes and link node delay time models. Fig. 4 is a typical networked delay system model. $V = \{v_i(s)\}$ represents the delay distribution function of the component node; $E = \{e_{ij}(s)\}$ represents the delay distribution function of the link node.

In control theory, if $V_1(s)$ is used as the input signal, $V_3(s)$ is used as the output signal. Then the expression between the input and output of the system can be expressed by the transfer function G(s):

$$G(s) = \frac{N(s)}{M(s)} = \frac{a_m s^m + a_{m-1} s^{m-1} + \cdots + a_1 s + a_0}{s^n + b_{n-1} s^{n-1} + \cdots + b_1 s + b_0} \tag{1}$$

where a and b are positive numbers and $m < n$.

Similarly, it is also possible to create a system that describes the delayed cumulative output of data after passing through multiple network nodes. If each node is represented as the delay distribution function of the node, the total transfer function from the input node to the output node is the delay cumulative distribution function from the source node to the destination node [18]. Therefore, as long as we know the delay distribution of data on each network node, the analysis based on the frequency domain method can produce an accurate delay time model. Moreover, an accurate delay time model can be obtained, and high-order characteristics of the delay are captured. Once the accurate delay time model is obtained, the mature theory in control engineering, such as root locus and Bode plot, is utilized. It is extremely convenient for analyzing network delay bottlenecks, and improving network real-time performance and reliability.

4. Cloud cooperation computing migration strategy selection

4.1. Cloud collaborative computing migration architecture

Cloud cooperative computing migration refers to an open platform that integrates network, computing, storage, and application core capabilities on the side close to the source of data or data itself, and provides services nearby. Its applications are launched on the edge side, thereby resulting in faster network service response, and meeting the industry's basic needs for real-time processing, smart applications, security and privacy protection. Cloud collaborative computing migration is between physical entities and industrial connections, or at the top of physical entities. In the cloud computing, historical data of cloud cooperative computing migration can still be accessed. If cloud computing is likened to the brain of the entire computer intelligence system, the cloud collaborative computing migration is the eyes, ears and hands of the system. The core server gives the intelligent system a strong artificial intelligence, but if the artificial intelligence is a scorpion, it will not play a big role. A common problem in big data applications is that no suitable data is collected. Cloud collaborative computing migration provides the most accurate and timely source of data for the core server's big data algorithms.

The combination of cloud collaborative computing migration and cloud computing makes the entire intelligent system clear-headed, smart and flexible. A computer system that relies entirely on cloud computing is like asking the army of the command for everything. When it takes a lot of interaction with the outside world, it will appear

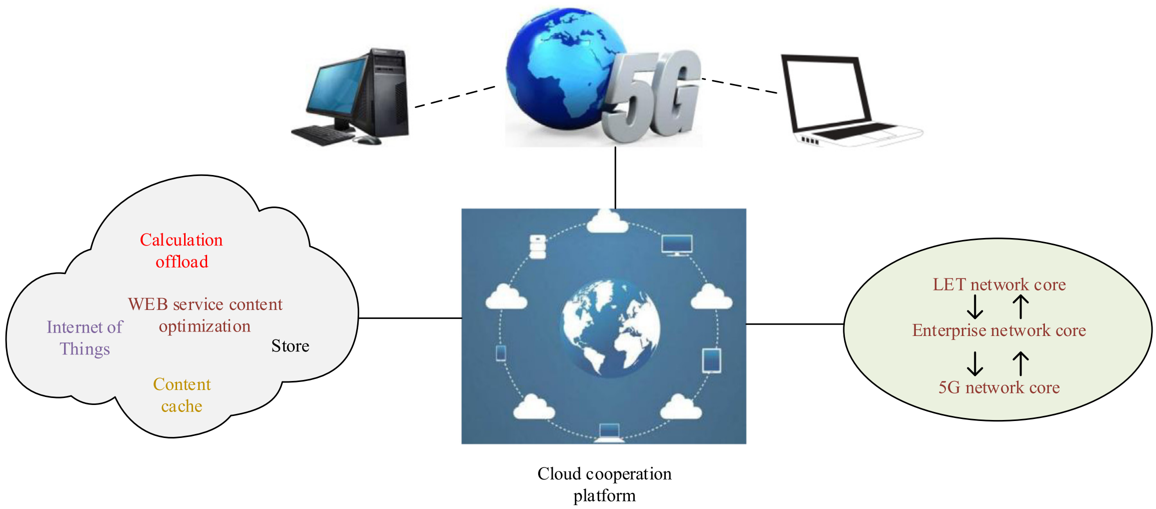

Fig. 5. Architecture of cloud collaborative computing migration.

rigid and slow to respond, and in the event of network problems, the system will be completely paralyzed. In addition, after the cloud cooperation calculation migration, it is like letting the middle and lower-level officers also begin to exert subjective initiative; and it can make intelligent judgments and action decisions to a certain extent [19–21]. At the same time, only a part of the filtered information needs to be uploaded to the headquarters, which greatly eases the pressure of network communication. Even if you temporarily lose contact with the headquarters, you can make some decisions on your own. The architecture and application diagram of cloud cooperative computing migration is shown in Fig. 5:

4.2. Calculating migration decision making

Cloud task migration mainly includes six steps: environment awareness, task partitioning, migration decision, task upload, edge server execution, and return result. Among them, task division and migration decision are two core steps. The specific flow chart of task migration is shown in Fig. 6:

Migration decisions are an important step in the task migration process. It refers to the process of using a certain indicator as an optimization goal, measuring the utility of the migration through relevant scientific methods, and making sub-tasks to the remote server. Task migration is usually divided into method-level migration, task-level migration, and application-level migration. Ordinary research is based on task-level migration. Task-level migration refers to the task-based granularity of dividing the application code into multiple parts and making partial modifications to ensure that each task can run independently. The mobile device uses an indicator to optimize the target, and feeds back the monitored bandwidth, the mobile device, and the parameters of the remote server's CPU and memory to the migration decision module. By calculation and measurement, the process of whether to uninstall and which subtasks to uninstall is made. Migration destination selection strategy based on supply and demand similarity and dynamic price model can ensure load balancing of cloud computing platform, improve resource utilization, and effectively reduce delay and user cost.

5. Optimal migration strategy selection based on task partitioning

In the traditional task migration strategy, the entire mobile terminal application is generally used as a migration object, or the task is divided into multiple subtasks of a chain-linear relationship before the migration decision is made. These do not take into account the complexities of multiple dependencies within mobile terminal applications. This paper will make a migration decision for mobile terminal applications with complex topological relationships, and obtain the migration decision result for each sub-task, which is the minimum delay migration scheme.

5.1. Linear task partitioning model

In the traditional task migration strategy, an entire mobile terminal application is often used as a migration object, and it is not divided into multiple subtasks. In this coarse-grained migration strategy, the low latency advantage of the mobile edge computing platform is not fully utilized, and the capabilities of the mobile edge computing platform are not considered to be lower than those of the cloud computing center.

The execution flow diagram of the fine-grained linear task migration scenario is shown in Fig. 7. It considers that each subtask is either executed on the mobile terminal side or on the mobile edge side. It records the calculation delay generated when the task is executed on the mobile terminal side as C(ME), and also calculates the calculation delay generated when the task is performed on the mobile edge side as C(EE). Moreover, when the execution location of a current subtask and the execution location of the latter subtask are different, a transmission delay occurs. The current person is executed at the mobile terminal, and when the latter is executed on the mobile edge side, a transmission delay and a transmission delay are generated, and the transmission delay is recorded as C (SID); the current performer is performed on the mobile edge side; and when the latter is executed on the mobile terminal side, a reception delay and a reception delay are generated, and the reception delay is recorded as C (ROD); it should be noted that the initial task data is stored on the mobile terminal side, and when all the subtasks are completed, the final processing result needs to be returned to the mobile terminal.

The above task partitioning model does not consider that the input data of a subtask may come from the output of multiple subtasks. It must be completed after these subtasks are complete.

5.2. Fine-grained directed acyclic graph-like task partitioning model

Fig. 8 is a topological model of a fine-grained directed acyclic graph. It uses a directed acyclic graph (DAG) to represent complex dependencies between subtasks. For Fig. 9, G = (V, E), where the node set V represents the set of tasks to be processed, and $v \in V$ represents the divided subtasks; E represents a set of directed edges, indicating the dependencies between tasks; $e_{uv} \in E$ represents the amount of data

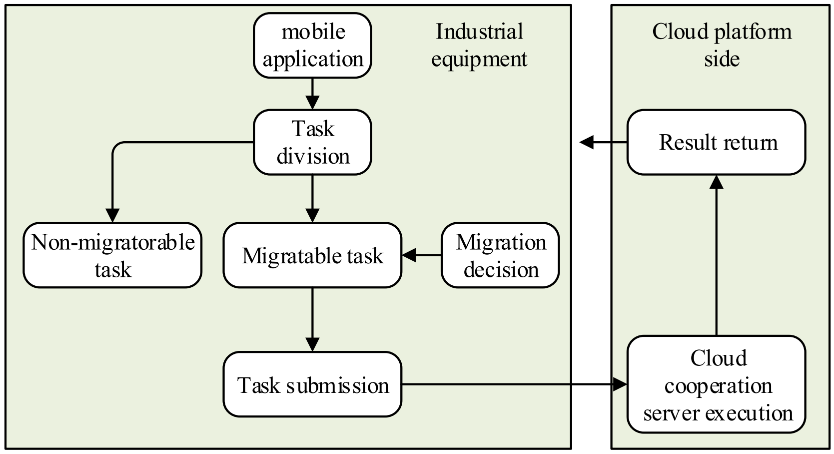

Fig. 6. Flow chart of task migration.

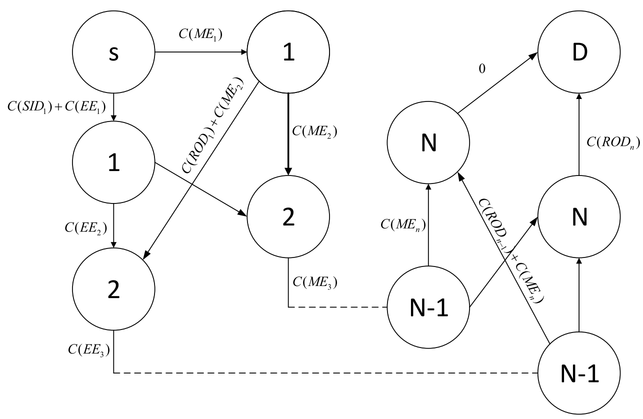

Fig. 7. Fine-grained status task execution process.

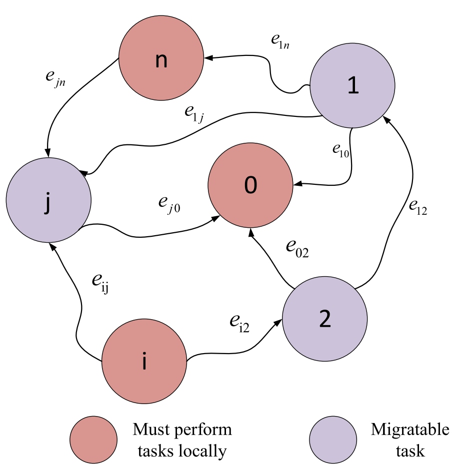

Fig. 8. Task topology model diagram with fine-grained directed acyclic graph.

transferred between task u and task v. It means that when the task u is executed, it will transmit the uve size data to the task v; task v execution can only be started after the task u has been executed and the data transmitted by the task u has been received.

5.3. Simulation condition

This article is based on Cloud Analyst software (CloudSim-based visual simulator, cloud computing simulation software developed by the *Buyya cloud* computing research team), which improves the main components by modifying the Cloud Analyst simulation platform code. This enables it to be transformed into a minimal delay simulation software for simulation experiments on the improved simulator.

The simulation scenarios in this chapter are as follows:

Within a certain geographical range, different network access points have different coverage ranges. Multiple mobile terminal users are also randomly distributed within the signal coverage of each network access point. In the simulation scenario, there are 4 mobile terminal users User1, User2, User3 and User4. They are in areas 2, 1, 3, 0; there are 4 network access points AP1, AP2, AP3, and AP4, which are located in areas 1, 2, 0, and 3. There are 9 MEC servers in this study, of which MECS7, MECS8 and MECS9 are on the AP3 side of the access point and are located in area 0; MECS4 is located on the AP1 side of the access point and is located in area 1; MECS1, MECS2, and MECS3 are on the AP2 side of the access point and are located in area 2; MECS5 and MECS6 are on the AP4 side of the access point and are located in area 3. The network access point configuration is shown in Table 2:

User configuration is shown in Table 3:

The delay matrix configuration is shown in Table 4:

Table 2
Configuration of network access points in this study.

Name	AP1	AP2	AP3	AP4
Region	1	3	2	0

Table 3
User configuration.

Name	User one	User two	User three	User four
Region	1	3	2	0
Request Per Hr	60	60	60	60
Date size per request	100	100	100	100

Table 4
Delay matrix (units in millisecond).

Region	0	1	2	3	4	5
0	50	100	150	250	250	150
1	100	50	200	500	300	200
2	125	200	50	150	200	300
3	150	500	150	50	400	500
4	250	350	200	400	50	300
5	200	250	250	500	500	50

5.4. Analysis of simulation results

In order to analyze the performance of the migration location selection strategy based on fine-grained directed acyclic graph-like tasks this

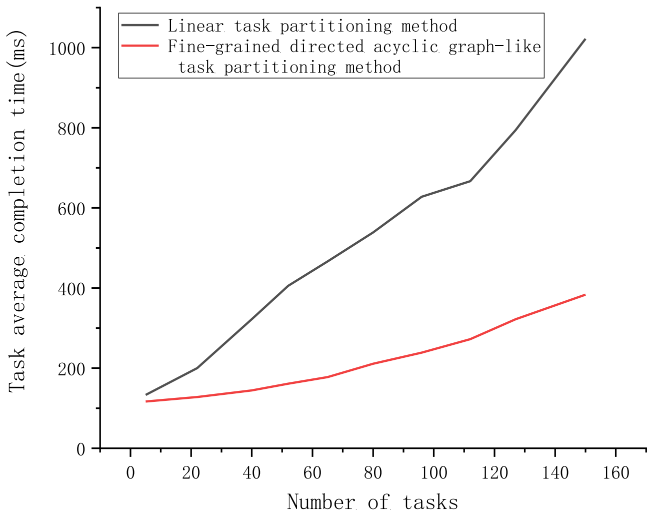

Fig. 9. Delay time comparison chart for different strategies.

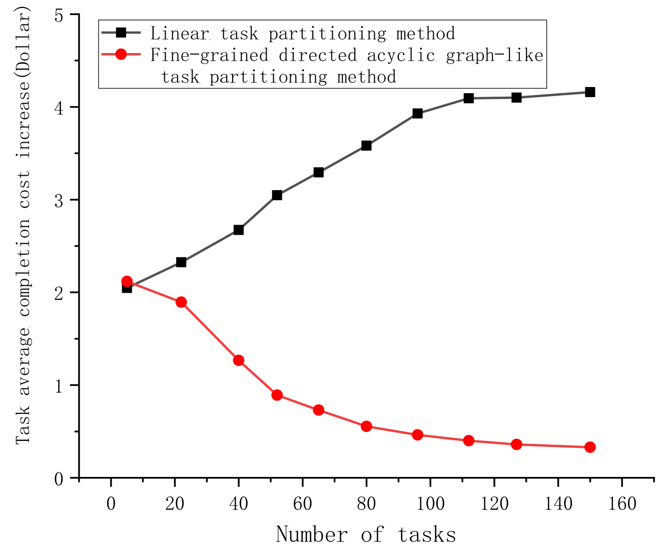

Fig. 10. Comparison of task costs for the three strategies.

chapter discusses that the linear task partitioning selection strategy is used as the comparison strategy. The task completion time comparison diagram of the two strategies is shown in Fig. 9. The figure shows the comparison of the average task completion time of the two strategies in the MEC environment with different number of application tasks participating in the migration. The task completion time is the time interval from the start of the task to the completion of the task and the return of the result to the user.

It can be seen from Fig. 9 that the fine-grained directed acyclic graph-like task partitioning strategy is always superior to the linear task partitioning strategy, and its task completion average time is always the smallest. When the number of migration tasks is small, the advantage of the fine-grained directed acyclic graph task division strategy on the average task completion time is not obvious. This is because the available resources at the beginning are sufficient; the resource utilization is relatively balanced; and the corresponding virtual machine can be deployed. As the number of tasks participating in the migration increases, the average task completion time is much lower than the linear task partitioning strategy. This is because the total amount of resources calculated by the mobile edge is fixed. When making the migration decision, the strategy fully considers the resource requirements of the mobile terminal application and the remaining amount of resources of the MEC server, and always migrates the task to the most suitable computing resource as much as possible, aiming to balance the resource utilization of the MEC server. This allows you to deploy more virtual machines to perform tasks with minimal latency.

Fig. 10 is a comparison of task overhead under two strategies. A comparison of the average mission overhead costs of the algorithms is shown in Fig. 10. The task overhead cost refers to the actual scheduling overhead cost paid by the user for completing the migration of a task in the mobile edge computing environment.

As can be seen from Fig. 10, as the number of tasks increases, the competition between tasks is fierce, and the task waiting time is longer. Therefore, the average completion time of the task increases; the unit price of the resource increases dynamically; and the average task cost increases. Based on fine-grained directed acyclic graph-like tasks, the average task migration overhead of a migration strategy is significantly better than the other. This is because the former makes the migration decision and takes into account the utilization and price of computing resources. It is always more inclined to migrate the task to a computing node with a relatively low price and a relatively low price. This reduces the migration overhead cost of application tasks to a certain extent.

6. Conclusions

The large capacity, large bandwidth, low latency, and low power consumption have gradually become the main demands of the industrial Internet of Things, with the rapid development of intelligent manufacturing. This paper proposes a fine-grained directed acyclic graph-like task partitioning strategy, abandoning the traditional stochastic selection idea, and proposes the concept of supply–demand similarity. Considering the similarity between the resource requirements of the task and the amount of computing node resources of the mobile edge computing platform, this study improves the resource utilization balance to ensure that more virtual machines can be deployed to perform tasks and perform simulation experiments under the CloudAnalyst simulation platform. The results show that the fine-grained directed acyclic graph-like task partitioning strategy is superior to the stochastic selection strategy in terms of task average completion time and delay, especially in the case of a large number of tasks. The IoT open platform provides access, storage and access services for data. However, the deep mining of data applications is far from enough. The specific business of BI (business intelligence) in enterprise information systems, such as decision support and predictive alarms, still needs data analysis system implementation. Providing intelligent services to users is the most important goal of the Internet of Things, big data mining is still a research direction for us in the future.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ke Wang: Conceptualization, Methodology, Writing - original draft.

References

[1] F. Tao, Y. Cheng, L. Da Xu, L. Zhang, B.H. Li, CCIoT-CMfg: cloud computing and internet of things-based cloud manufacturing service system, IEEE Trans. Ind. Inf. 10 (2) (2014) 1435–1442.

[2] F. Tao, Y. Zuo, L. Da Xu, L. Zhang, IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing, IEEE Trans. Ind. Inf. 10 (2) (2014) 1547–1557.

[3] H. Topcuoglu, S. Hariri, Min-You Wu, Performance-effective and low-complexity task scheduling for heterogeneous computing, IEEE Trans. Parallel Distrib. Syst. 13 (3) (2002) 260–274.

[4] Shafii Muhammad Abdulhamid, Muhammad Shafie Abd Latiff, Syed Hamid Hussain Madni, Mohammed Abdullahi, Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm, Neural Comput. Appl. 29 (1) (2018) 279–293.

[5] Wonsub Lim, Dongil Yang, Youngoo Yang, An antenna proximity sensor for mobile terminals using reflection coefficient, Sensors 18 (7) (2018) 2103.

[6] Anup Das, Francky Catthoor, Andre Bourdoux, Bert Gyselinckx, Energy efficient mapping of lte-a phy signal processing tasks on microservers, IEEE Trans. Green Commun. Netw. 2 (2) (2018) 397–407.

[7] Luís Ferreira, Goran Putnik, Maria Manuela Cruz-Cunha, Zlata Putnik, Leonilde Varela, A cloud-based architecture with embedded pragmatics renderer for ubiquitous and cloud manufacturing, Int. J. Comput. Integr. Manuf. 30 (4) (2017) 483–500.

[8] L. Maglaras, L. Shu, A. Maglaras, J. Jiang, H. Janicke, D. Katsaros, T.J. Cruz, Industrial internet of things (I2oT), Mob. Netw. Appl. 23 (4) (2018) 806–808.

[9] S.G. Pease, R. Trueman, C. Davies, J. Grosberg, K.H. Yau, N. Kaur, A. West, An intelligent real-time cyber-physical toolset for energy and process prediction and optimisation in the future industrial Internet of Things, Future Gener. Comput. Syst. 79 (2018) 815–829.

[10] J. Wan, S. Tang, Z. Shu, D. Li, S. Wang, M. Imran, A.V. Vasilakos, Software-defined industrial internet of things in the context of industry 4.0, IEEE Sens. J. 16 (20) (2016) 7373–7380.

[11] M. Liu, J. Ma, L. Lin, M. Ge, Q. Wang, C. Liu, Intelligent assembly system for mechanical products and key technology based on internet of things, J. Intell. Manuf. 28 (2) (2017) 271–299.

[12] Z. Bi, L. Da Xu, C. Wang, Internet of things for enterprise systems of modern manufacturing, IEEE Trans. Ind. Inform. 10 (2) (2014) 1537–1546.

[13] D. Bandyopadhyay, J. Sen, Internet of things: Applications and challenges in technology and standardization, Wirel. Pers. Commun. 58 (1) (2011) 49–69.

[14] L. Zhang, Y. Luo, F. Tao, B.H. Li, L. Ren, X. Zhang, Y. Liu, Cloud manufacturing: a new manufacturing paradigm, Enterp. Inf. Syst. 8 (2) (2014) 167–187.

[15] R.Y. Zhong, X. Xu, E. Klotz, S.T. Newman, Intelligent manufacturing in the context of industry 4.0: a review, Engineering 3 (5) (2017) 616–630.

[16] Y. Bai, Industrial internet of things over tactile internet in the context of intelligent manufacturing, Cluster Comput. 21 (1) (2018) 869–877.

[17] M.C. Chen, S.Q. Lu, Q.L. Liu, Global regularity for a 2d model of electro-kinetic fluid in a bounded domain, Acta Math. Appl. Sin. Engl. Ser. 34 (2) (2018) 398–403.

[18] Y. Cheng, F. Tao, L. Xu, D. Zhao, Advanced manufacturing systems: supply–demand matching of manufacturing resource based on complex networks and Internet of Things, Enterp. Inf. Syst. 12 (7) (2018) 780–797.

[19] Imene Djelloul, Zaki Sari, Mehdi Souier, Fault isolation in manufacturing systems based on learning algorithm and fuzzy rule selection, Neural Comput. Appl. 31 (8) (2019) 3211–3225.

[20] O. Zimmermann, Architectural refactoring for the cloud: a decision-centric view on cloud migration, Computing 99 (2) (2017) 1–17.

[21] M. Ellison, R. Calinescu, R.F. Paige, Evaluating cloud database migration options using workload models, J. Cloud Comput. 7 (1) (2018) 6.