Intelligent warehouse task information flow scheduling under fog computing environment

Chen Lei¹,a, Jia Minzhi¹,b
¹Taiyuan University of Technology, College of Electrical and Power Engineering, Taiyuan, Shanxi, China
aemail: chenlei5068@163.com, bemail: tutjmz666@sina.com

Abstract: Traditional cloud computing systems used in intelligent warehouses are managed in a centralized manner. Faced with more and more intelligent terminal devices in the warehouse, the resources of the intelligent warehouse cannot be fully utilized. Therefore, a distributed approach is adopted to deploy fog computing nodes near smart terminal devices to provide real-time computing response. The network architecture of intelligent warehouse based on fog computing is designed, and a real-time task scheduling model is established considering the time delay problem. In this model, an improved artificial bee colony algorithm is used to schedule the task information flow and allocate a reasonable amount of computing data to the fog nodes. The simulation results show that the error between the calculated delay and the calibrated delay on the fog node is within the specified range and can meet the requirements of different task types for the time delay.

1. Introduction

With the development of intelligent warehouse, intelligent terminal equipment and sensors in intelligent warehouse are becoming more and more popular [1]. There is no need for manpower in the whole logistics process from storage to transportation of products. Based on the Internet of Things, the whole logistics process becomes more and more efficient and intelligent. Advances in the Internet of Things, robots, and drones have also eased the constraints of manual handling and conveyors.

Typically, the cloud in an intelligent logistics center is managed centrally. However, if the factory is too large and has too many tasks, the response time will be long and it will not be able to respond to the requests of the large intelligent terminal devices to the centralized cloud. Fog computing adopts a distributed mode. It takes the fog computing resources near the intelligent terminal to undertake the calculation tasks of logistics operation and make real-time response. These tasks can be completed more directly and flexibly, thus becoming a method to solve heavy load and high delay [2]. This paper analyzes the characteristics of intelligent warehouse, combines the advantages of fog computing, and adopts a network architecture of intelligent warehouse under fog computing environment. On the basis of this architecture, the task information flow is assigned to the fog nodes by the improved artificial bee colony algorithm, and the optimal amount of computing data is determined for the fog nodes.

2. The network architecture

Combined with the advantages of fog computing and the characteristics of intelligent warehouse itself, a new network architecture of intelligent warehouse is constructed. The whole network architecture can be divided into three layers: cloud computing layer, fog computing layer and terminal equipment layer, as shown in Figure 1.
Terminal equipment layer is composed of shuttle, AGV, stacker, camera, mechanical arm, RGV and other warehouse equipment and various sensors. The real-time status data required for decision is identified and collected by intelligent terminal devices and sensors, and then transmits a fog node device that temporarily stores and processes data, and then sends decisions back to the terminal device. At the same time, the data is also transmitted to the cloud computing center for historical analysis and long-term storage.

![Figure 1 Intelligent warehouse network architecture based on fog computing](image)

Fog computing layer is composed of several fog nodes and forms a specific fog network. The fog nodes in the fog computing layer collect data from the edges of low-level devices and respond and analyze the data in a short time. The fog node is closely deployed with all kinds of physical machines and terminal equipment in the intelligent warehouse, and provides real-time computing services through network connection. The fog node connects to the cloud computing layer and acts between the terminal facility and the cloud computing layer. The fog computing handles the high latency of cloud computing. The fog node has a strong heterogeneity. It can process the data uploaded by various devices, make quick decisions and upload the results to the clouds. It can also execute decisions on the cloud layer.

Cloud computing layer, consisting of high-performance large servers. The cloud computing layer provides a computing model that can be used to access the network anytime, and take advantage of shared computing tools, storage devices, and applications based on request and convenience. The cloud computing layer provides remote services for the intelligent warehouse. After the data tasks are transferred to the cloud center by the fog node, the cloud computing layer is responsible for processing the heterogeneous data in the warehouse with large computational complexity, high task level and large data volume through high-performance computing equipment and large-capacity storage equipment, and conveys decisions to the fog computing layer.

3. The model and problem formulation
The essence of the resource scheduling problem of fog computing is that, in the fog computing environment, intelligent terminal devices make requests, and the fog computing server processes these service requests in a timely and efficient manner according to the resource scheduling rules formulated. The goal of resource scheduling in fog computing environment is to meet the balance requirement of terminal equipment for time delay. The design goal is to make full use of fog computing resources to achieve task information flow distribution and minimize task processing time.

3.1. Task model
The terminal device layer puts forward a request to the fog computing dispatching center and submits the related tasks, and then the dispatching center matches the corresponding computing resources for it. In the architecture of fog computing, a target task to be completed can be decomposed into corresponding subtasks, which can run independently. The task set $M$ submitted by users consists of $p$ elements and can be expressed as $M=\{m_1, m_2, \cdots, m_i, \cdots, m_p\}$, where $m_i (i=1, 2, 3, \cdots, p)$ represents the $i$th
subtask and \( p \) represents the number of tasks.

### 3.2. Scheduling model

Near the intelligent terminal device layer, a series of virtual resources are used to process tasks, which can be called fog nodes. The set of fog nodes can be expressed as \( F = \{ F_1, F_2, \cdots, F_j, \cdots, F_q \} \), where \( F_j \) represents the \( j \)th fog node, and \( p > q \).

For intelligent warehouses, the terminal equipment is different, and the fog computing tasks to be handled and the fog nodes to be arranged are all heterogeneous. Therefore, the scheduling problem of fog computing resources is complex. The relation matrix is used to represent the scheduling scheme between the fog computing task and the fog node, and each subtask is mapped to the corresponding fog node through the corresponding mapping method. The relation between the task and the fog node is shown in matrix \( X \).

\[
X = \begin{bmatrix}
x_{11} & x_{12} & \cdots & x_{1q} \\
x_{21} & x_{22} & \cdots & x_{2q} \\
\vdots & \vdots & \ddots & \vdots \\
x_{p1} & x_{p2} & \cdots & x_{pq}
\end{bmatrix}
\]

Among them, \( x_j \) represents the affiliation between the task \( m_i \) and the fog node \( F_j \), \( x_j \in \{0,1\} \), \( \sum_{j=1}^{q} x_j = 1 \), \( i=1,2,3,\cdots,p \). When the task \( m_i \) calculates on the fog node \( F_j \), \( x_j = 1 \), otherwise \( x_j = 0 \).

If a differentiated scheduling scheme is taken, the fog calculation completion task is different. In order to analyze the data in time, the resource scheduling target function is to be established, the optimization goal is to make the fog calculation task to complete the shortest time. The time delay of each task is shown in the matrix \( T_I \) in the fog node:

\[
T_I = \begin{bmatrix}
t_{i1} & t_{i2} & \cdots & t_{iq} \\
t_{i1} & t_{i2} & \cdots & t_{iq} \\
\vdots & \vdots & \ddots & \vdots \\
t_{i1} & t_{i2} & \cdots & t_{iq}
\end{bmatrix}
\]

\( t_{ij} \) indicates the calculation task \( m_i \) calculation time delay on the fog node \( F_j \), \( t_{ij} = D_i \theta_i / C_j \), \( D_i \) represents the amount of data in the task information flow, \( \theta_i \) represents the density of task information flow, \( C_j \) represents the computing power of fog node \( F_j \). So the task completion time on the fog node:

\[
FT[j] = \sum_{i=1}^{p} T_{ij} * x_j
\]

Total time to complete the task:

\[
AT = \max_{j=1}^{q}(FT[j])
\]

### 4. Task scheduling algorithm design

An improved artificial bee colony algorithm is used for task information flow scheduling.

#### 4.1. Basic principles of artificial bee colony algorithm

Artificial Bee Colony Algorithm (ABC) is a bionic intelligent algorithm which simulates bee colony searching for nectar source \(^7\). In ABC algorithm, artificial bee swarm consists of three parts: employed foragers, unemployed foragers and scout bees. It mainly includes the following five stages:

1. Initialization phase. Where the population number of bees is \( N_c \) and the number of employed foragers is \( N_c / 2 \), the initial solution can be generated according to formula (5):

   \[
   \alpha_j = \alpha_{\min j} + \text{rand}(0,1)(\alpha_{\max j} - \alpha_{\min j})
   \]

   \( \alpha_j \) represents the jth dimension value of the ith solution, \( \alpha_{\min j} \) , \( \alpha_{\max j} \) represent the maximum and minimum values of the jth dimensional search space, \( 1 \leq i \leq N_c / 2 \), \( j \in \{1,2,\cdots,D\} \), \( D \) is the dimension of the
solution, rand(0,1) is a random number between 0 and 1.

2) Employed foragers use greedy selection method to compare the nectar amount of two nectar sources and selects the nectar source with more nectar amount. The neighborhood search formula is expressed as follows:

\[ \nu_j = \alpha_j + \text{rand}(-1,1)(\alpha_j - \alpha_k) \]  

\( \alpha_j \) represents the original nectar source location, \( \nu_j \) represents the new nectar source location, and \( \alpha_k \) is another nectar source different from the current nectar source. \( k \neq i \) and \( k \in [1, N_c] \).

3) According to the information brought back by the employed foragers, if the nectar amount is large, the probability of the unemployed foragers choosing the nectar source is large, and the nectar amount of the nectar source reflects the fitness of the solution. The probability formula related to the amount of honey from the nectar source is shown in formula (7):

\[ p_i = \frac{\text{fit}(i)}{\sum_{i=1}^{N_c} \text{fit}(i)} \]  

\( \text{fit}(i) \) is the fitness value of the ith solution, which is proportional to the amount of honey source at the ith position. The expression is as follows:

\[ \text{fit}(i) = \begin{cases} 
1, & \text{if} \text{fit}(i) \geq 0 \\
1 + |\text{fit}(i)|, & \text{if} \text{fit}(i) < 0 
\end{cases} \]  

(4) If a nectar source stays more than the limit number of times, and still cannot find a higher nectar source, give up. Next, a new nectar source is generated according to formula (5).

(5) After completing the above steps and reaching the termination requirements, the optimal solution reaches the goal.

4.2 Improved artificial bee colony algorithm
Reverse learning is added to artificial bee colony algorithm, the main idea of reverse learning is to refer to its corresponding reverse estimation points while selecting the estimation points, so as to get a better candidate solution. After the reverse learning is added, the reverse population is generated, which is combined with the forward population to make the population evenly distributed, reduce the occurrence of local optimal, and find the global optimal solution easily. The formula of reverse nectar source is:

\[ \alpha_j^r = \text{rand}(0,1)(\alpha_{\text{max}} + \alpha_{\text{max}}) - \alpha_j \]  

\( \alpha_j^r \) is the nectar source in the reverse nectar source and \( \alpha_j \) is the nectar source in the original nectar source. The reverse honey source and the forward honey source were searched simultaneously, and the search results were compared to select the optimal solution that was more in line with the requirements. In other words, the optimal solution obtained by forward search and reverse search is optimized again.

5. Experiment and result analysis

5.1 Parameter Settings
In order to prove the application of the reverse artificial bee colony algorithm in the task schedule, MATLAB is used as an experimental platform. Under the intelligent warehouse network architecture in the fog calculation environment, the number of fog nodes is 8, the number of intelligent terminals is 40. For selected fog nodes, the calculate ability is generally fixed; however, different intelligent terminals generate the calculation task, the amount of data is not necessarily the same, and the calculation density is different from the computing task.

It is known that the data volume of computing task generated by a terminal in the intelligent warehouse is 1~10MB, and the calculation density is 100~300cycles/b. The computing capacity of fog node in intelligent warehouse is 2Gcycle/s. When the type of computing task is known, the computational density is known. After the fog node is selected, the computing power is known.
Therefore, the calculation delay of selected fog nodes is controlled by the data volume of the task information flow. Because different terminals produce different types of computing tasks and different amount of task data, the resulting computing delay is not the same. However, terminal equipment has requirements on time delay, and different terminals produce different tasks, corresponding to different calibration time delay requirements. The delay can meet the requirements only if the amount of calculation data allocated by fog nodes is reasonable.

5.2. Results analysis
Because different task types correspond to different computational densities, the time delay requirements are different. When the amount of data distributed by the fog node is reasonable, the difference between the calculation delay and the calibration delay is smaller, so as to meet the delay requirements.

The calculation density of task 1 is 300cycles/b, the calculation capacity of fog node is 2Gcycle/s, and the calibration delay is 0.5s. After the artificial bee colony algorithm with reverse learning is added, it can be seen from Figure 2 that the algorithm can find the optimal solution after about 13 iterations, so that the difference between the calculation delay and the calibration delay is approximately equal to 0. In this case, the convergence rate is fast enough and the calculation delay meets the requirements.

Figure 2 The relationship between the number of iterations and the calculation result (Task one)
As shown in Fig. 2, for task 1, when the assigned data volume of fog node is 3.332MB, the calculated delay basically meets the task requirements. The data amount allocated by the fog node is within the corresponding range of the task data amount generated by the terminal.

The calculation density of task 2 is 200cycles/b, the calculation capacity of fog node is 2Gcycle/s, and the calibration delay is 0.9s. After the artificial bee colony algorithm with reverse learning is added, it can be seen from Fig. 3 that the algorithm can find the optimal solution after about 10 iterations, so that the difference between the calculation delay and the calibration delay is approximately equal to 0. In this case, the convergence rate is fast enough and the calculation delay meets the requirements.

Figure 3 The relationship between the number of iterations and the calculation result (Task two)
As shown in Fig. 5, for task 2, when the assigned data volume of fog node is 9.0MB, the calculated delay basically meets the task requirements. The data amount allocated by the fog node is within the corresponding range of the task data amount generated by the terminal.
In the intelligent warehouse, the selected fog nodes are determined, and the computing capacity of the fog nodes is also determined. For different types of computing tasks, the difference in computational density and the difference in the amount of computing data will lead to different delay requirements. Through simulation, different computing tasks can obtain different computing delays. While finding the optimal data allocation of fog nodes, the absolute error between calculation delay and calibration delay is less than 0.1%. The algorithm adopted in this paper meets the requirements, as shown in Table 1:

| Calculation of the density (cycles/b) | Ability to calculate (Gcycles/s) | Calibration time delay (s) | Optimum data volume (Mb) | Calculate time delay (s) | Error (%) |
|--------------------------------------|---------------------------------|---------------------------|--------------------------|-------------------------|-----------|
| 1                                    | 300                             | 2.000                     | 0.500                    | 3.332                   | 0.4998    | 0.04      |
| 2                                    | 300                             | 2.000                     | 0.900                    | 6.002                   | 0.9003    | 0.03      |
| 3                                    | 200                             | 2.000                     | 0.900                    | 9.000                   | 0.9000    | 0         |
| 4                                    | 200                             | 2.000                     | 0.500                    | 4.999                   | 0.4999    | 0.02      |

6. Conclusions
Terminal equipment and sensors in intelligent warehouse produce a large number of computing tasks. In order to meet the requirements of computing delay, the network architecture of intelligent warehouse in fog computing environment is designed. An improved artificial bee colony algorithm is used to schedule the task information flow of fog nodes, and a reasonable amount of computing data is allocated to the fog nodes. Experimental results show that the proposed scheduling model and algorithm can meet the time delay requirements of computing tasks. This paper innovatively explores the application of fog computing in intelligent warehouse, and then continues to study the location selection and computing capacity selection of fog nodes.

Reference
[1] Kayikci, Y. (2018) Sustainability impact of digitization in logistics. Procedia Manufacturing, 21, 782–789.
[2] Sarkar, S., Misra, S. (2016) Theoretical modelling of fog computing: a green computing paradigm to support IoT applications. IET Networks, 5(2): 23–29
[3] Balevi, E., Gitlin, R. D. (2018) Optimizing the Number of Fog Nodes for Cloud-Fog-Thing Networks. IEEE Access, 6, 11173–11183.
[4] Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., et al. (2010) A view of cloud computing. Communications of the ACM, 53(4), 50.
[5] Wan, J., Chen, B., Wang, S., Xia, M., Li, D., Liu, C. (2018). Fog Computing for Energy-aware Load Balancing and Scheduling in Smart Factory. IEEE Transactions on Industrial Informatics, 14:4548–4556
[6] Yang, Y., Zhao, S., Zhang, W., Chen, Y., Luo, X., Wang, J. (2018) DEBTS: Delay Energy Balanced Task Scheduling in Homogeneous Fog Networks. IEEE Internet of Things Journal, 5(3), 2094–2106.
[7] Karaboga, D., Gorkemli, B. (2011) A combinatorial Artificial Bee Colony algorithm for traveling salesman problem. In: 2011 International Symposium on Innovations in Intelligent Systems and Applications. Istanbul, Turkey. pp. 50-53