Journal of Electronic Research and Application

OPEN

An Efficient Task Scheduling Method for the Unified Interface Platform of the Electric Information Acquisition System

Ye Fangbin, Han Xiaohan, Wang Chaoliang, Tian Jiale

State Grid Zhejiang Electric Power Research Institute, Hangzhou 310014 chaoliangwang@126.com, yefb@163.com, tianjiale53@s163.com

Abstract: Due to the large and frequent static data interaction between the Electric Information Acquisition System and the external business systems, researching on using limited server sources to do an efficient task scheduling is becoming one of the key technologies of the unified interface platform. The information interaction structure of the unified interface platform is introduced. Task scheduling has been decomposed into two stages, task decomposition and task combination, based on the features (various types and dispersed) of large static data. The principle of the minimum variance of the subtasks data quantity is used to do the target task resolving in the decomposition stage. The thought of the Greedy Algorithm is used in the task combination. Breaking the target task with large static data into serval composed tasks with roughly same data quantity is effectively realized. Meanwhile, to avoid the situation of the GA falling into the local optimal solution, an improved combination method has been put forward. Moreover, the new method creates more average composed tasks and making the task scheduling more effective. Ultimately, the effectiveness of the proposed method is verified by the experimental data.

Key words: Electric information acquisition system; unified interface platform; task decomposition; Greedy Algorithm (GA)

Published Online: 31st Jan 2018

Introduction

With the Electric Information Acquisition system ("acquisition system") full coverage, full collection, other business systems have more demands on the acquisition system data [1-4], as of August 2016, the collection system has provided business and data support for marketing, safety and quality, transport inspection, transport supervision, policy deliver, information and many other external business systems, involving 16 national power grid companies' integrated business application system and dozens of provincial electric power companies' self-built business application system, and the collection system is becoming more and more prominent as the power data support platform for each business system of electric company [1-6].

In order to meet the requirements of other external business systems, to avoid the development of the same interface and to reduce the pressure of operation and maintenance, a set of unified interface platform of the electric information acquisition system ("unified interface platform") is built to interact with the external business system. Because of the wide interactive task and large amount of data, it has been one of the key technologies to study the efficient task scheduling with limited server resources. At present, the unified interface platform mainly carries on four kinds of data interaction including the static data interaction, the real-time data interaction, the process type data interaction and the external data to obtain, and the static data interactive data amount occupying 85% of all interactive tasks, is the most major data interaction form in the unified interface platform. The static data of the unified interface platform has the characteristics of large volume and dispersed type, so it is necessary to

study the high efficiency scheduling technology of the static data interaction task of the unified interface platform in order to meet the requirement of fast and high efficiency processing for the related service of electric information collection.

For the unified interface platform static data interaction, this paper divides task scheduling into 2 phases: task decomposition and task combination. In the phase of decomposition, the task is decomposed by the minimum principle of the amount variance of the subtask data, and the greedy algorithm [7-9] is used in the task combination to realize the large static data task decomposed into a set task with roughly consistent data volume. At the same time, in order to avoid the problem that greedy algorithm may fall into local optimal solution, an improved combination method is proposed, which makes the set task more average, improves the efficiency of server thread pool, and finally realizes the task of unified interface platform efficiently.

1. Information interaction structure of unified Interface platform

The unified interface platform of the electric information acquisition system is divided into 5 layers: interaction type, interactive information, task control, information processing and data model. Interaction types include four obtaining forms of static data interaction, real-time data interaction, process interaction and external data acquisition; interactive information includes five types of static request object, static return object, real-time publishing object, interactive data object and external data object; task control mainly includes request analysis and task scheduling; information processing mainly includes data processing and file generation; data model includes load data, electricity data, power quality, terminal working condition, index statistic data, event data, process interaction data, external data and so on. The overall structure the information interaction for the unified interface platform is shown in Fig. 1.

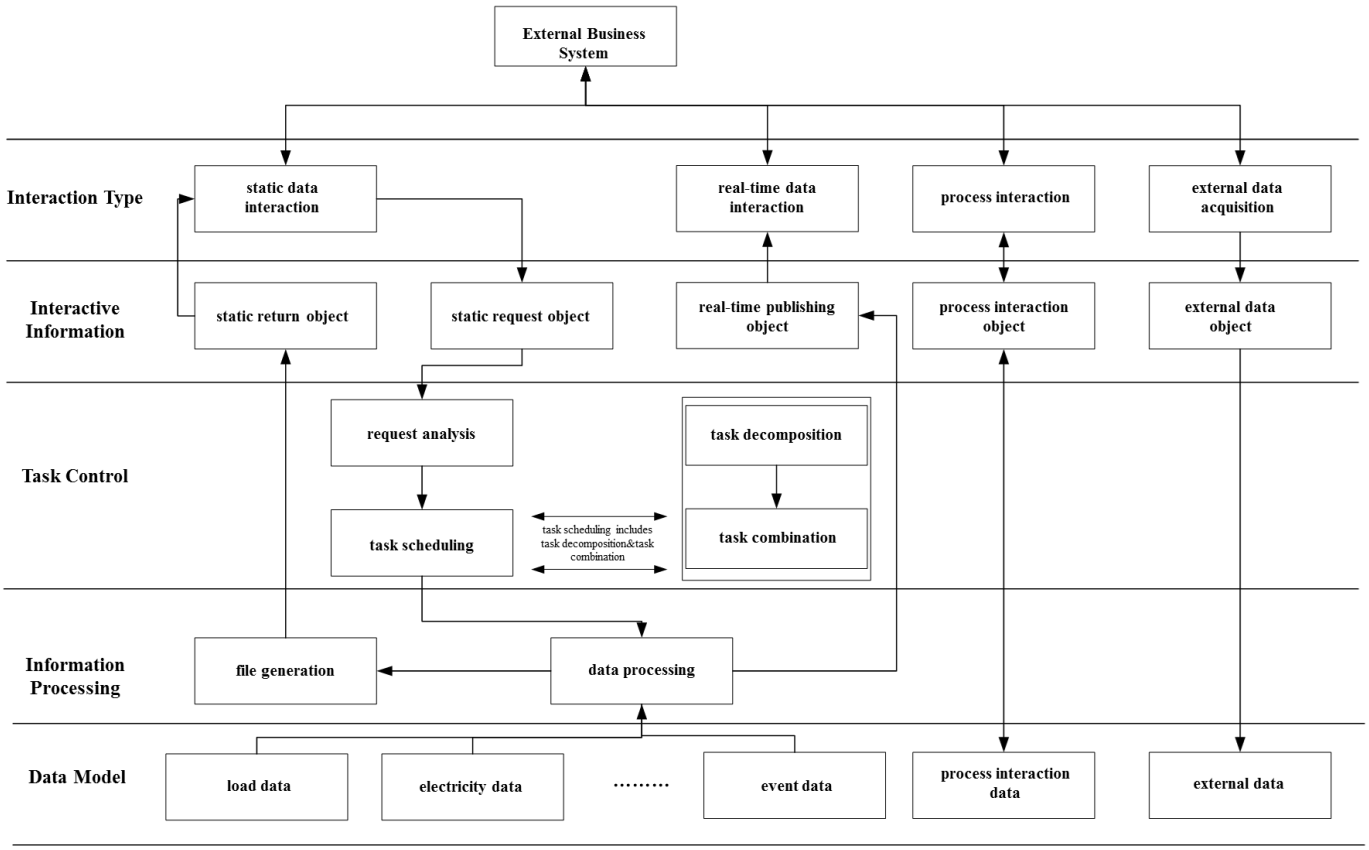

Figure 1. Structure of the information interaction for the unified interface platform

The task scheduling is the core algorithm of task control, which needs to be optimized on a stable and reasonable basis to improve efficiency.

Unified interface Platform provides business and data support for marketing, safety and quality, transport inspection, transport supervision, policy deliver,

information and many other professional business application systems, the provided data containing electricity, demand, load curve, all kinds of statistical data and other static data, terminal events, meter events and other real-time data, information, work orders and process information and other process interaction data. For the actual work demand, the static data interactive data account for more than 85% of all interactive tasks, which is the most important data interactive form of the unified interface platform. Therefore, this paper will take the static data as main research object. The static data provided to external business systems by the unified interface platform are as shown in tab. 1.

No.	Data type	Data scope	Data frequency
1	display	City, County, institute	day
2	electricity	City, County, institute	day
3	load	City, County, institute	Day, hour, minute
4	Load characteristic	City, County, institute	day
5	demand	City, County, institute	day
6	anomaly	City, County, institute	day
7	Working condition	City, County, institute	all
8	index	City, County, institute	day
9	Voltage statistics	City, County, institute	day

Table 1. Static interactive data of the unified interface platform

The static data of the unified interface platform is analyzed from the aspects of business characteristics, data characteristics and aging characteristics, as follows:

（1）Analysis of business characteristics

Business involves the user scope analysis, the user scope includes: specially-changed users, commonly-changed users and the Low-voltage users;

Business involves the granularity analysis, the data granularity includes: province, city, county, institute, district, users;

Business entity object Analysis, entity object includes: line, transformer, user, terminal, meter.

（2）Analysis of data characteristics

Data content analysis, the data content includes: electricity, load, events, anomaly, statistical indicators;

Data frequency analysis, the data frequency includes: quarter, month, week, day, hour, minute, real-time.

（3）Analysis of aging characteristics

According to the interaction interval analysis, the interaction interval includes: monthly, daily, quasi real-time, real-time;

According to the time range analysis, the time range includes: single point of time, time period.

The analysis shows that the static data types of the unified interface platform are numerous and dispersed, which can be divided into several dimensions, providing the basis for the multithread task scheduling of the unified interface platform.

For large static data task scheduling, in the case of a certain number of threads in the server, the purpose of task scheduling is to achieve the average distribution of large static data tasks to all threads, that is, large static data task as far as possible with the average allocation of data as N set tasks, to achieve efficient use of threads. So this paper divides task scheduling into 2 interpretations: task decomposition and task scheduling.

Task decomposition: The target task is divided according to the dimension (such as date, user range, etc.), and the P dimension is decomposed, there are 2^p decomposition methods altogether.

Task combination: Assuming that the number of existing threads is N, the total task is decomposed into n parts by task decomposition. In order to make full use of each thread's resources and improve query efficiency, it plans to combine the n small tasks of the decomposed task into N tasks, and make the data of the N tasks close to each other, that is, each thread completes the assigned task as close as possible, making the total processing time (that is, the time all threads complete the task) the shortest.

2. An efficient scheduling method for tasks

The task scheduling of unified interface platform requires the optimal decomposition and combination of the target task under the limited number of threads. In the task decomposition phase, the target task should be decomposed into as many subtasks as possible; in the task combination phase, the subtasks can be combined

into a set task with uniform data volume as much as possible. The more subtasks the target task breaks down during the decomposition phase, the more times the server accesses in the combined stage, and the more the corresponding resource consumption increases. Therefore, the unified interface platform to achieve efficient task scheduling, needs to make a trade-off between efficiency and resource loss, in this paper, based on greedy algorithm, an efficient task scheduling method is proposed. Suppose the target task is R, the total amount of data is S, the decomposition dimension is P, and the total number of server threads is N, $S(i)$ indicates the amount of data for task i. By decomposing and combining, the task R is divided into N set tasks with approximately equal data amount, which can achieve efficient dispatch, see in Fig. 2, as follows:

Step one: The decomposition phase, use P dimensions to decompose the target task R, a total of 2^p decomposition methods, the subtask number decomposed by the target task is $n \in (\delta_1 N, \delta_2 N)$ condition decomposition method is defined as a feasible decomposition method, where δ_1 , δ_2 （$1<\delta_1<\delta_2$）according to the actual situation (server performance, etc.) to set. Make m_i to represent the decomposition method i （$i \in \{1,2,3 \ldots 2^p\}$） to decompose the target task to get the number of subtasks, $\{x(t)|t=1,2,\ldots x\}$ represents a set of feasible decomposition methods.

For all feasible decomposition methods $i \in \{x(t)|t=1,2,\ldots x\}$, the variance of subproject data after decomposition by different decomposition methods is calculated, and is recorded as σ_i. By comparing the variance size of the subtasks of all feasible decomposition methods, the decomposition method represented by min of variance minimum $(\sigma_i)_{min}$ is the best decomposition method. The aim of this algorithm is to select the decomposition method that the number of subtasks satisfies the $n \in (\delta_1 N, \delta_2 N)$ and the minimum variance of subtask data. In the case of the same number of subtasks, the more average the data size of the subtasks, the more favorable to the balanced distribution in the combined stage.

Step two: The combination stage, after the completion of the target task decomposition, combine subtasks based on the greedy algorithm, so that the subtask data are distributed into the average combination of N set tasks as much as possible. Make $S(k)$to represent the current assignment to the collection task k（$k=1,2,\ldots N$）, which initializes $S(k)=0$ for all collection tasks.

According to the amount of data, arranges subtasks from large to small, and they are numbered $j=1,2,\ldots,n$. Assuming that the data quantity of the subtask j is S_j, the data quantity of the assignment sub task j is assigned to the set task with the smallest data quantity τ, and the above operation is repeated until all the subtasks are assigned, and a collection task of N data quantity is formed to complete the task combination.

Through step one and step two, the unified interface platform will decompose and combine the target task into N set tasks, corresponding to the server existing N threads, avoiding thread idle causing resource waste, improving task scheduling efficiency.

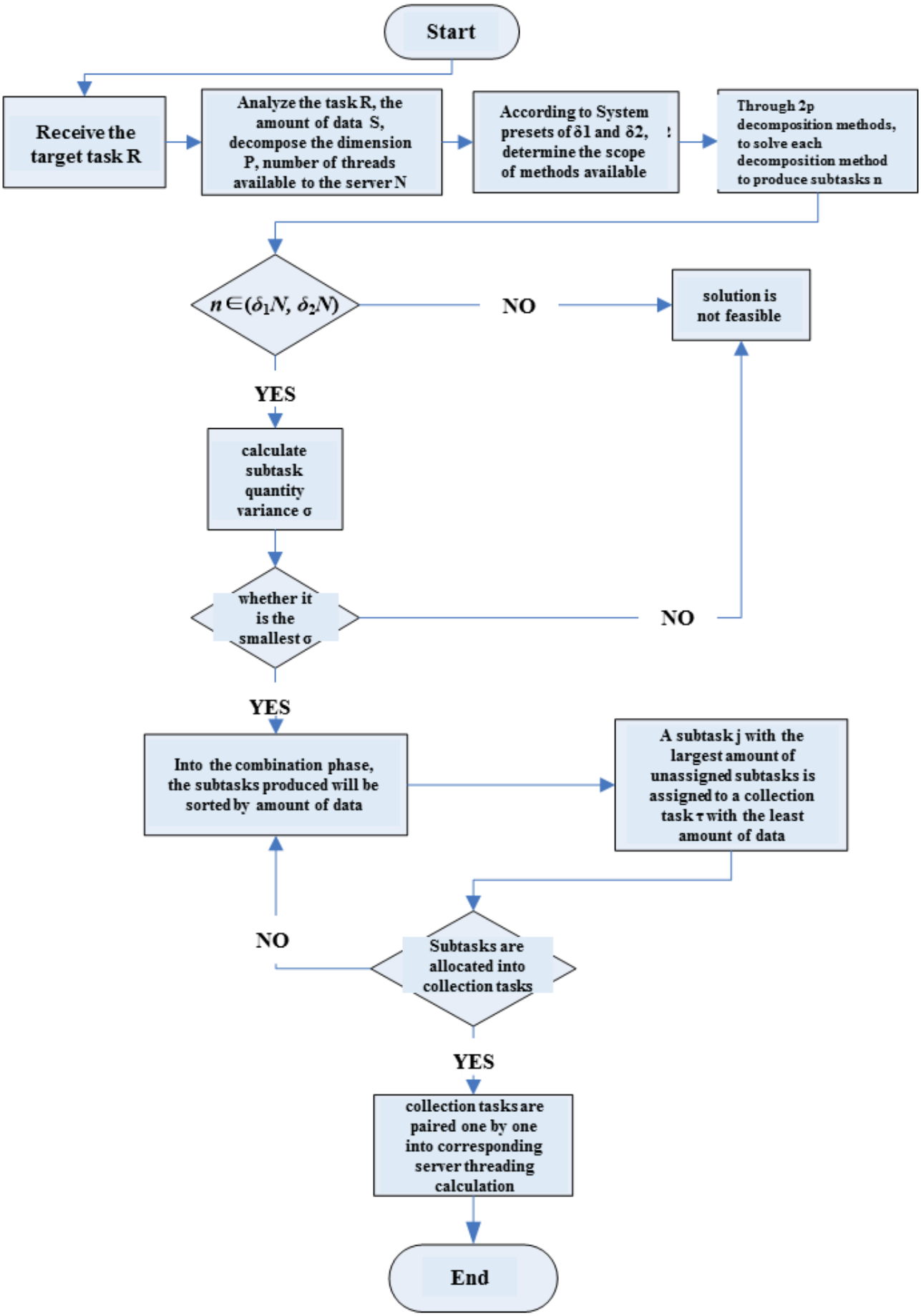

Figure 2. Flow chart of the efficient task scheduling

3. An improved combination method

Because the greedy algorithm may fall into the local optimal solution, the combination method using the greedy algorithm alone may not achieve the optimal result [7-9]. The combination method in step two can be adjusted as follows: by exchanging the data amount of the largest difference between the two task subtasks, the data volume of each collection task can be more evenly balanced.

Suppose the set task k has an f subtasks, numbered 1,2,3,.., f, which defines Φ_k as all of the subtasks that can be exchanged to other collection tasks on the collection task k. Therefore, the following exchange rules are designed:

(1) For set task i and set task j, suppose $S(i) > S(j)$, $\mu_{ij}=(S(i)- S(j))/2$;

(2) Look for subtasks y_i、y_j, which, $y_i \in \Phi_i$、$y_j \in \Phi_j$, make $0<S(y_i)-S(y_j)\leq\mu_{ij}$, while the $S(x_i)-S(x_j)$ value is the largest.

Repeatedly repeating the above rules allows the collection task i and the collection task j to be as close as possible to the amount of data. When you select the collection task i and the set task j, you should optimize the two set tasks with the largest difference in the current amount of data. If there is no solution to the two set tasks with the largest difference in data volume, then the two set tasks with the second largest amount of data are computed, and so on. The solution termination condition is that for any two set task i and Set task j, they cannot find the corresponding y_i, y_j to satisfy $0<S(y_i)-S(y_j)\leq\mu_{ij}$, as shown in Figure3.

Through the improvement of the Two combination method in 3rd step, a more efficient scheduling method can be obtained.

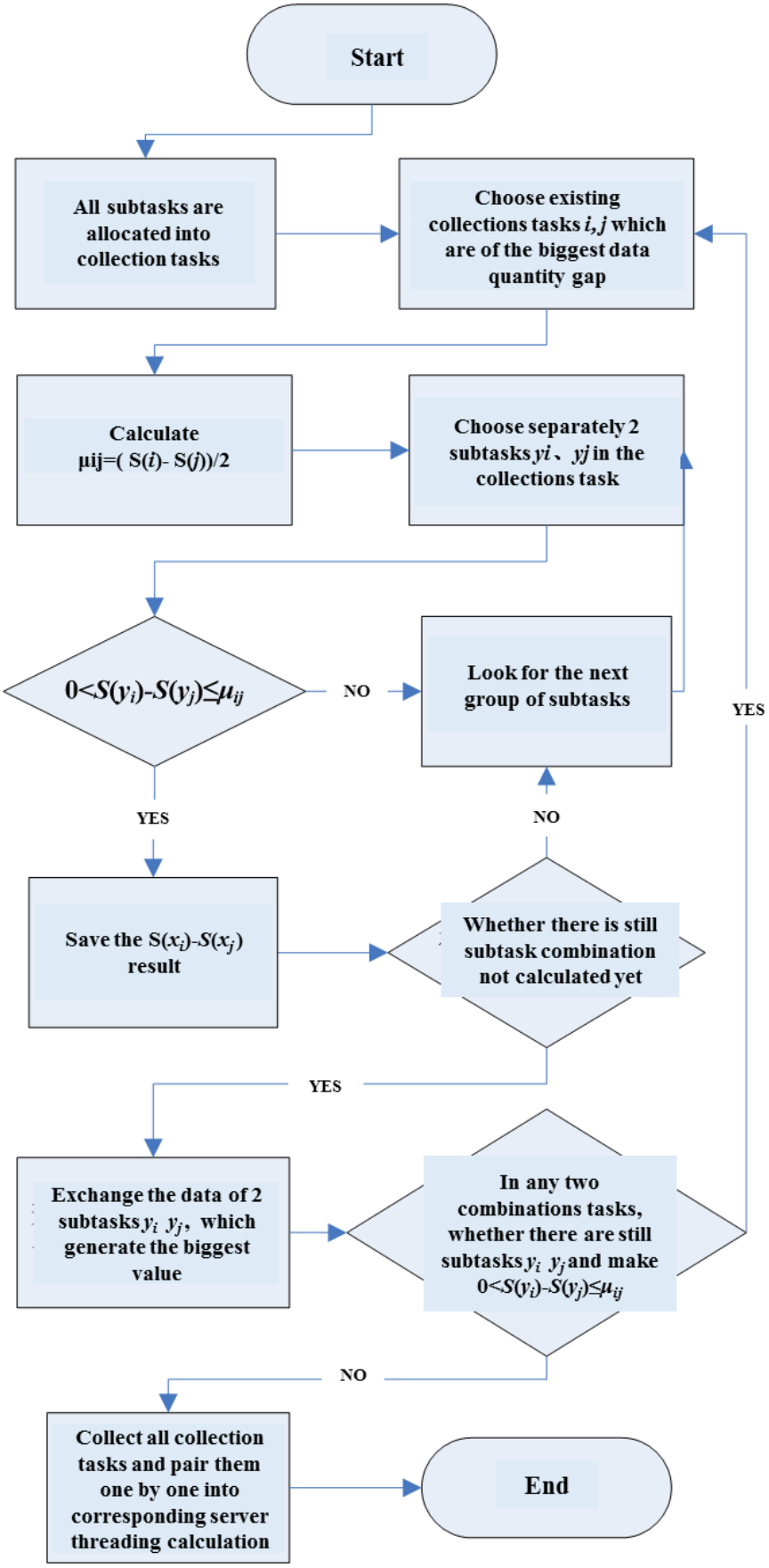

Figure 3. Flow chart of the improved combination method

4. Experimental verification

In order to verify the effectiveness of the efficient scheduling method proposed in this paper, taking

electricity information data of a certain province electric power company as an example, the data quantity of the unified interface platform dealing with large static data target task ("large task") is 1000G, the thread available for server has N=10 pieces altogether, the decomposition dimension has 4 kinds, respectively named dimension *A*. *B*, *C*, *D*. Each dimension can decompose a large task into multiple subtasks, and the different dimensions of the larger task are decomposed as shown in table 2.

Dimension	Decompose subtasks（unit：G）
A（6）	100,150,120,200,250,180
B（4）	250,175,225,350
C（8）	120,100,80,135,150,175,75,165
D（5）	140,150,210,240,260

Table 2. Results of the decomposition based on different dimensions

$P(q)$indicates the number q of subtasks obtained by the dimension P decomposition large task, and it can be found in table 2 that different dimensions can be decomposed into different subtasks.

Step one: Decomposition phase

Through the combination of 4 dimensions of *A*. *B*, *C*, *D*, a total of 16 decomposition methods are available, set the parameters δ_1=3, δ_2=5, then meet that 4 feasible decomposition methods are available when the number of subtasks is $m \in (\delta_1 N, \delta_2 N)$ after decomposition, as shown in table 3.

Decomposition method	Decomposition subtasks（unit：G）
AD（30）	43,46,10,23,34,22,60,38,37,26,36,39,35,16,35,23,48, 51,14,38,34,30, 56,40,29,28,20,53,14,22
BC（32）	25,32,52,21,8,18,32,18,3,54,50,58,23,11,37,15,30,28, 4,23,50,46,55, 31,62,38,28,42,16,7,53,30
CD（40）	23,8,30,31,35,43,31,47,42,4,7,45,8,34,36,11,40,27,11, 25,21,32,7,28,17,32,38,6,47,12,27,7,40,4,18,22,20,8, 25,51
AC（48）	43,10,2,20,30,12,30,30,13,11,18,13,30,18,36,16,12,14,11,8,14,6,7,32,20,6,33,19,19,24,9, 7,38,12,30,22,42,39,31,3,30,30,14,3,40,23,41,29

Table 3. Results of the feasible decomposition methods

Calculate the subtask variance of 4 feasible decomposition methods:

var1(AD)=163.0889

var2(BC)=281.0625

var3(CD)=190.0000

var4(AC)=137.2639

Since var4(AC) is minimal, AC（48）is chosen as the best decomposition method for large task decomposition, with a total of 48 subtasks.

Step two: Combination phase

According to the step two algorithm, the 48 subtasks obtained in step one are grouped into 10 subtasks, allowing the task to run simultaneously in 10 threads, and the collection tasks obtained by step two are shown in table 4.

thread	Collection Tasks (unit: G)	The amount of data (unit: G)
1	43,23,19,11,7	103
2	42,24,19,11,6	102
3	41,29,14,12,6	102
4	40,30,14,12,3	99
5	39,30,16,10,8	103
6	38,30,18,9,7	102
7	36,30,18,12,3	99
8	33,30,20,13,2	98
9	32,30,20,14	96
10	31,30,22,13	96

Table 4. Results of the greedy algorithm combination method

The first column represents the thread label, the second column represents the combination of tasks for each thread, and the third column represents the amount of data for each collection task. Through step one and step two, the collection task data volume of each thread is very close, which proves the effectiveness of the efficient scheduling method.

For the results obtained by the greedy algorithm in step two, we use the improved combinatorial method to fine tune, the results as shown in table 5.

Thread	Collection tasks (unit: G)	The amount of data (unit: G)
1	43,20,19,11,7	100
2	42,22,19,11,6	100
3	41,29,13,12,6	101
4	40,30,16,12,3	101
5	39,30,14,10,8	101
6	38,30,18,9,7	102

7	36,30,18,12,3	99
8	33,30,20,14,2	99
9	32,30,23,14	99
10	31,30,24,13	98

Table 5. Results of the improved combination method

Compared with the results obtained by the greedy algorithm in step two, after fine-tune of the improved combination method, the collection task data on each thread are more average, which improves the efficiency of the whole operation.

6. Conclusions

Aiming at the problem of large static data task scheduling for the unified interface platform of electric information acquisition system, this paper presents a task efficient scheduling method, which divides task scheduling into 2 phases: task decomposition and task combination. In the phase of decomposition, the task is decomposed by the minimum principle of the amount variance of the subtask data, and the idea of the greedy algorithm is used to carry out the subtask combination in the task combination, which can effectively decompose the large static data task into a set task with roughly consistent data volume. At the same time, in order to avoid the problem that greedy algorithm may fall into local optimal solution, an improved combination method is proposed, which makes the set task more average, improves the efficiency of server thread pool, and finally realizes the task of unified interface platform efficiently.

References

[1] Hu Jiangyi, Zhu Enguo, Du Xingang, Du Shuwei. Application Status and Development Trend of Power Consumption Information Collection System [J]. Automation of Electric Power Systems, 2014, (02): 131- 135.

[2] Yuwen Xiaodi. Research and Application of Information Collection System for Electricity Users [D]. North China Electric Power University (Beijing), 2011.

[3] Chen Sheng, Lv Min. Power Usage Information Acquisition System and Its Application [J]. Distribution & Utilization, 2011, (04): 45- 49.

[4] Zhang Xujie. Design and Implementation of Electric Information Collection System [D]. North China Electric Power University,2013.

[5] Chen Fei. Research on Information Security Interaction Model and Key Technology of Intelligent Power Grid [D]. North China Electric Power University (Beijing), 2014.

[6] Zhang Jing, Lu Jiepin. Research and Implementation of Information Interaction Network Model in Small and Medium Cities [J].

INDUSTRIAL CONTROL COMPUTER, 2001, (10): 46- 48 +61.

[7] Wang Youzhao, Peng Xuxiang, Pan Fenlan. Scheduling algorithm for vehicles in warehouses based on greedy algorithm and genetic algorithm [J]. Transducer and Microsystem Technologies, 2012, (10): 125- 128.

[8] Raja Giryes. A greedy algorithm for the analysis transform domain[J]. Neurocomputing, 2016,173.

[9] Dorigo M. Ant Colony Optimization [M]. Scholarpedia, 2007.