Cloud Platform Task Scheduling Strategy for Power Dispatching Automation System Based on Artificial Intelligence Algorithm

Shiming Li¹, Bo Li¹, Ke Wang¹, Wenxin Guo¹, Yang Liu¹, Ruiyin Zhao¹, Jianyong Zeng¹, Haobin Li¹, Yuehong Lin¹
¹Electric Power Dispatching Control Center, Guangdong Power Grid Corporation, Guangzhou 510600, China

Abstract: As the command and control center of the whole power grid operation, the power dispatching system has huge and complex data information, which makes the traditional power grid dispatching information platform difficult to meet the needs of power grid operation monitoring in information integration, analysis, calculation and storage. Therefore, cloud platform is introduced into the construction of power dispatching automation system. This paper takes the power dispatching automation cloud platform as the research object, studies its architecture, and proposes a cloud platform task scheduling model based on improved particle swarm optimization, aiming to improve the efficiency of cloud computing resource allocation, improve the quality of cloud services, and solve the task scheduling problem of the power dispatching automation system.

1. Introduction
The cloud computing platform of the intelligent power grid dispatching system based on the interconnection of the internal power grid has a strong ability of data collection, analysis and calculation. The task scheduling of the power dispatching automation cloud platform is to use the computer server cluster to distribute the software and hardware resources of the whole power grid stably and effectively, so that the overall resource consumption is less and the execution efficiency is high when the system executes and processes tasks.

Power dispatching automation system cloud platform is a shared pool that supports users to obtain computing resources through the network, and the shared pool can be managed to optimize the business model of rapid resource configuration and delivery. The cloud platform utilizes modern Internet technology to provide massive scalable and flexible IT resources as a paid service to a large number of cloud users. Cloud users do not need to care about the service implementation process when acquiring services, but only need to pay attention to their own service needs. The resources in the cloud platform are dynamic, while the needs of cloud users are real-time, diverse, and dynamic. Therefore, pre-configuration and fixed resource allocation have been unable to meet the service needs of cloud users.

In power dispatching automation system cloud platform, task scheduling is the first principle to meet the needs of users, with the goal of optimal task scheduling scheme and maximum overall benefit. This requires that the cloud platform task scheduling process must have flexibility, real-time and efficiency, so that the massive resources of cloud computing can efficiently meet the needs of cloud users. However, traditional task allocation and task scheduling methods cannot meet the service quality required by cloud platform users for power dispatching automation system, and it is difficult to ensure cloud platform load balancing. Therefore, efficient and scientific dynamic resource scheduling algorithm and resource
allocation technology in cloud computing for power dispatching automation system have become the key to improving cloud computing user experience and cloud platform usage efficiency. This paper proposes a cloud platform task scheduling model for power dispatching automation system based on improved particle swarm optimization, aiming to improve the efficiency of cloud computing resource allocation, improve the quality of cloud services, and solve the task scheduling problem of power dispatching automation system.

2. Connotation and Objectives of Cloud Platform Task Scheduling

2.1 Connotation

Scheduling problems, especially the task scheduling problems in multiprocessors and distributed systems, have always been the research focus of computer science. At this stage, with the development of cloud computing, task scheduling under the cloud platform has also been favoured by many scholars, and more and more researchers have invested in the task scheduling research under cloud computing. Cloud platform task scheduling can be understood as a mapping process. Through virtualization technology, the software and hardware facilities of the cloud platform are abstracted as resources, and the user's request is regarded as a task. The role of task scheduling is to allocate necessary and appropriate cloud resources to perform tasks for users' submitted tasks based on the status and prediction information of both resources and tasks under certain constraints. After the task is completed, the processing result is returned and resources are released. Therefore, a good task scheduling mechanism can coordinate and allocate resources in the cloud environment, effectively improve system performance, shorten task running time, and reduce resource consumption.

There are two basic starting points for cloud platform task scheduling: on the one hand, transparently allocate resources for tasks submitted by users; on the other hand, through a certain algorithm scheduling, the resources in the cloud and the tasks submitted by the user are matched with the maximum utilization rate, the utilization efficiency of the resources in the cloud is improved, and the user is provided with better and more efficient services. Cloud computing is the inheritance and development of distributed computing and grid computing, so it has a larger scale and complexity, and the resources in the cloud environment are heterogeneous, dynamic, distributed, and autonomous. Therefore, for cloud platforms, a good and efficient scheduling strategy is essential.

The task scheduling process of cloud computing is very similar to the task scheduling process of grid computing. Cloud platform task scheduling model is shown in Figure 1.

```
Figure 1. Cloud platform task scheduling model
```

In general, the execution process of cloud platform task scheduling is as follows:

a) Decompose the application according to the application configuration file;
b) Resource selection based on resource pool and resource pool configuration file;
c) Perform task mapping and calculate the resources required by the task;
d) Assign resources to the tasks submitted by users based on resource configuration files and resource pool usage.

Task scheduling problem is an NP-hard problem, because resources and tasks are dynamically changing, which is not conducive to task scheduling. Therefore, the scheduling algorithm can only find a relatively optimal solution to the task scheduling problem, and the scheduling service quality QoS is used as the evaluation standard.

2.2 Objectives

The objectives of cloud platform task scheduling include the following aspects:

a) Shortest span. The span refers to the completion time of the task, that is, the time it takes from the cloud computing system to execute the first task submitted by the user to the end of the last task. The length of the span directly affects the user experience. The shorter the span, the less time it takes, and the less the user needs to pay, and this is what the user expects.

b) Quality of service (QoS). Service quality refers to the pros and cons of the performance of the system as reflected by the cloud computing platform when providing services. The level of QoS directly determines whether users will continue to use the cloud computing platform. QoS in the cloud platform is mainly reflected in the performance of task scheduling, that is, the allocation of virtual resources. In a heterogeneous cloud computing platform, how to efficiently schedule virtual resources is the key to improving service quality.

c) Load balancing. Load balancing is a key indicator for measuring system resource scheduling in cloud computing, which refers to the balanced use of virtual resources under the cloud computing platform to avoid the situation where some nodes are overloaded and some nodes have no tasks to do. In order to achieve this effect, it is necessary to use a reasonable task scheduling strategy to balance the relationship between resources and tasks while making full and reasonable use of virtual resources.

d) Economic benefit. Cloud computing is a pay-as-you-go service, so for service providers and users, cost is a factor that needs to be balanced. For users, reducing service fees is what they want; while for service providers, it is desirable to reduce costs while ensuring overall revenue.

3. Architecture of Cloud Platform for Power Dispatching Automation System

The intelligent power dispatching automation system uses cloud computing technology to integrate distributed IT data information resources and power grid automation infrastructure in a structured manner through a distributed data service bus to form a high reliability, high real-time, high-accuracy smart grid dispatch cloud platform.

The power dispatching automation system cloud platform takes distributed data service bus as the core component, including dynamic load balancing and resource allocation system, distributed massive data storage system and integrated computing engine. These functional components form the virtual layer through the distributed data service bus, which realizes the exchange, transmission and integration of control information and data information. At the same time, they manage and deploy the underlying physical hardware in a unified way, which provides a guarantee for various applications to be called and accessed efficiently and stably.

In order to facilitate the scheduling system administrator to monitor the use and operation status of system components in real time, and to monitor, adjust and schedule them in real time, the platform also has a unified management and monitoring interface. The basic cloud platform of each dispatching center in different regions of the smart grid has these three functional components. The sub dispatching center in each region can be integrated into the network interconnection communication platform through the distributed data service bus. The functional components of different dispatching centers have the function of mutual backup, which improves the safety and reliability of the system.

3.1 Distributed data service bus

Power dispatching system cloud platform is a large-scale distributed networking architecture with flexible network topology. Multi-level dynamic scheduling can be carried out among the sub scheduling
platforms. The bus can integrate the servers of each sub dispatching platform organically, so that the underlying business resources and the function of low delay data transmission can be shared among the nodes, which provides a guarantee for the continuous availability of the system. The distributed data service bus also supports modifying configuration parameters, adding dynamic applications, protocol expansion, etc., and ensures the automatic recovery of the system after node failure and process failure. Its efficient remote management function and powerful log function make it convenient for dispatchers to adjust system parameters, monitor network conditions, and provide reliable guarantee for finding and managing the system.

3.2 Distributed mass data storage system
Distributed mass data storage system is a high-performance software system that can integrate a large number of independent storage nodes on the network to solve the massive real-time data in power operation scheduling. Without adding new equipment and changing the physical location of hardware, it can deal with the storage and management of massive data in the long-term business development process of power grid by using distributed data management technology.

Distributed mass data storage system puts the data to be processed on the bottom distributed file system in the form of files, and builds a virtualization system on the local file system of all database systems in the cluster. Through the dynamic division and management of data nodes in each database system, the data synchronization and exchange between multiple database systems can be processed at the same time, and the distributed database storage also supports the storage architecture of multiple masters and slave coexisting. Through the construction of unstructured peer-to-peer cluster of multiple master-slave nodes, the difficulty of data access is solved, and the speed of data processing in power dispatching system is improved.

3.3 Dynamic load balancing and resource allocation system
In order to effectively solve the management problem of power dispatching system, the cloud platform integrates the dynamic load balancing and resource allocation mechanism. The dynamic load balancing and resource allocation mechanism dynamically allocates tasks according to the computing speed of each computing node device through the central control point in the system, and deploys the load balancing server of each node in the whole network at one time, monitors the information of important nodes and pressure load in real time, which can make the pressure load of the whole network dynamically adjusted by the system. The four main functions of central monitoring system, certification management center, strategy management center and decision center are provided by the management center of the platform. The central monitoring system is responsible for the load situation of the whole power grid; the certification management center and the strategy management center are responsible for the certification and management of the working nodes of the whole power grid. Decision center makes a global judgment on power operation and dispatching system.

When the system monitors the failure of some nodes, it adopts the mechanism of fault tolerance and task redistribution. The nodes with low load in the whole network are found first, and the tasks of the failed nodes are transferred to these low load nodes for calculation.

3.4 Integrated computing engine
Many kinds of computing resources are integrated into this computing engine, and the massive data processing problems generated in the process of task scheduling are solved in a hierarchical way, which effectively improves the real-time processing and analysis of large-scale data of the system. This kind of computing engine can submit the massive data information collected by the intelligent terminal step by step from the external and internal nodes, as shown in Figure 2. The analysis and processing of data information are carried out by different function nodes one by one until all the data collected by the intelligent terminal is completed.
Figure 2. Integrated computing engine

4. Cloud Platform Task Scheduling Model for Power Dispatching Automation System Based on Artificial Intelligence Algorithm

The cloud platform task scheduling strategy based on artificial intelligence algorithm can be described as: cloud platform users send application requests to the cloud computing data center, and the data center allocates virtual resources according to the system policy according to the request, and maps the optimal physical resources through the improved particle swarm algorithm, and then responds to user requests and provide users with corresponding services. There are two core links in this task scheduling process: one is Virtual machine allocation strategy, which is formulated according to user needs, specifically based on constraints; the other is physical machine mapping, which uses an improved particle swarm algorithm to optimally map virtual machines to physical machines to achieve resource scheduling.

This paper uses an improved particle swarm artificial intelligence algorithm to build a cloud platform task scheduling model. We combine the improved particle swarm algorithm model with the network topology of the cloud computing resource scheduling model, and regard cloud computing to treat physical machine resources in task scheduling as particles in particle swarm algorithm, and the entire physical machine resource pool as particle swarm, so as to solve the problem of cloud platform optimization task scheduling. Cloud computing physical machine resource information corresponds to particle information in improved particle swarm optimization. While the communication ability of the physical machine corresponds to the speed of the particles, and the physical machine resources with strong processing ability correspond to the individual optimal particles in the particle swarm. The task assignment process in cloud computing corresponds to the particle elimination process in the improved particle swarm optimization algorithm, and the constraint conditions for task assignment correspond to the particle selection conditions in the improved particle swarm optimization algorithm. In the cloud environment, the improved particle swarm optimization algorithm performs iterative updates based on the predicted completion time and the actual situation of resource responsibility to obtain the optimal resource scheduling solution. In this scheduling scheme, the QoS function value is combined with the utility function to evaluate the merits of the optimal scheduling scheme, and at the same time, the penalty value can be set by the penalty function to optimize the scheduling scheme.

4.1 Model construction

In order to apply the improved particle swarm optimization algorithm to the task scheduling of cloud platform, the first step is to build the mathematical model of task scheduling of cloud platform. Our work is based on the specific model and principle of cloud computing resource scheduling, establish practical objective function, determine the specific constraints in the scheduling process, and set the
relevant parameters are the basis of the improved particle swarm optimization algorithm applied to cloud computing resource scheduling.

A good task scheduling in the cloud platform can establish a balance between the user's needs and the load balance of the cloud environment. Cloud platform users are eager to obtain timely and reliable services at low cost, while cloud platform providers hope to provide stable services at the condition of using less resources and low energy consumption, so that limited resources can create the greatest commercial value. Therefore, the amount of resources used and the quality of service of resources in cloud computing environment determine this balance. It can be seen that the more resources there are, the more resources cloud computing users can choose, and the more likely they are to get timely, reliable and economic services. However, with the increase of resources, the more complex the cloud computing environment is, the more difficult it is to ensure the stability of the system. Therefore, we build an optimal task scheduling model that can balance user needs and service expectations as follow:

$$F = Q(p) + G(p)$$

The mathematical image of this model is shown as Figure 3.

![Figure 3. Optimal task scheduling model](image)

In the formula, $Q(p)$ represents the relationship between resource quantity and user service expectation; $G(p)$ represents the relationship between the number of resources and the load balancing of cloud environment. The balance between the two is to find the minimum value of $F$, the corresponding resource quantity is the resource quantity that can be used when the two can be balanced, that is, the value of the graph intersection.

(1) Objective function

Using artificial intelligence algorithm to solve the optimal task scheduling problem is a conditional task scheduling process according to the service requirements of cloud users under the condition of considering load balancing. The resource data corresponding to the optimal task scheduling is the solution of the objective function, which turns the problem into the problem of finding the optimal task scheduling. When selecting a physical machine, the upper limit of its load rate is considered, which ensures the load balance of the cloud environment from the source. We can establish the objective function as follows:

$$F = \min(h_1 \times \frac{\text{finiTime}}{\text{time}_{\text{pre}}} + h_2 \times \frac{\text{cost}}{\text{cost}_{\text{pre}}} + h_3 \times \frac{\text{trust}_{\text{pre}}}{\text{Trust}(U)})$$

In the function, $\text{time}_{\text{pre}}$, $\text{cost}_{\text{pre}}$ and $\text{trust}_{\text{pre}}$ respectively represent cloud users' reservation requirements for service time, service fee and service reliability.

(2) Constraints

Before task scheduling, it is necessary to calculate the fitness value of the task, eliminate some tasks according to the fitness value of the task, and then optimize the allocation of tasks from the remaining tasks according to the constraints. In this process, specific constraints need to be considered. The constraints set of the objective function is as follow:
\[
\begin{align*}
\text{Delay}(VM_i) & \leq DL \\
\text{Time}_{\text{cost}}(VM_i) & \leq TL \\
\text{Bandwidth}(VM_i) & \leq BL \\
\text{Trust}(VM_i) & \geq TrL \\
\text{fit}(VM_i) & \geq \varepsilon \\
\text{finiTime}(U) + \text{Delay}(U) & \leq \text{time}_{\text{pre}} \\
\text{total Cost}(U) & \leq \text{cost}_{\text{pre}} \\
\text{Trust}(U) & \geq \text{trust}_{\text{pre}}
\end{align*}
\]

In the formula, \(\text{Time}_{\text{cost}}(VM_i)\) represents the time cost required by the virtual machine \(VM_i\) to process the task, the upper limit is \(TL\); \(\text{Delay}(VM_i)\) represents the time delay required by the task to the virtual machine \(VM_i\), the upper limit is \(DL\); \(\text{Bandwidth}(VM_i)\) represents the maximum network bandwidth can provide when tasks are assigned to \(VM_i\), the lower limit is \(BL\); \(\text{Trust}(VM_i)\) represents the trusted value of the virtual machine \(VM_i\), the lower limit is \(TrL\); \(\text{fit}(VM_i)\) is the fitness of the task; \(\varepsilon\) is the minimum fitness value of the selected task.

(3) Parameter setting

According to the principle of task scheduling in cloud platform based on improved particle swarm optimization algorithm, relevant parameters need to be set. The parameter setting of the improved particle swarm optimization algorithm in this paper is shown in Table 1.

| Parameters | Meaning | Range of values |
|------------|---------|----------------|
| \(\eta\)  | Inertia coefficient | (1.5, 2.5) |
| \(\alpha\) | Individual learning factors | (0, 4) |
| \(\beta\)  | Group learning factors  | (0, 4) |
| \(\lambda, \mu\) | Random number  | [0, 1] |
| \(V_{\text{max}}\) | Maximum particle velocity  | 100 |
| \(\delta\) | Maximum constraint function value | 2 |
| \(I_{\text{max}}\) | Maximum number of iterations | 1000 |
| \(C_1\)  | Precocity determination constant | 1 |
| \(C_2\) | Adjustment coefficient of inertia factor | [0.85, 0.95] |
| \(e\) | Upper limit of physical machine load rate | [0.7, 0.8] |

4.2 Implementation process

According to the structure model of cloud platform task scheduling optimization and the established objective functions and constraints, the implementation steps of improved particle swarm optimization algorithm in cloud computing resource scheduling are as follows:

Step 1: Taking the service requirements of cloud computing users as the objective function to guide the particle search of cloud computing task scheduling of improved particle swarm optimization;

Step 2: Determine the parameters of the algorithm as Table 1, according to the size of the problem, and Set the position and speed of the initialization particle, and make the initialization particle optimal;

Step 3: Calculate the initial fitness value of particles according to the fitness function of particles, and eliminate some particles according to the elimination rules;

Step 4: Update the particle swarm information according to the iterative rules, and update the individual optimal \(P_{\text{best}}\) and group optimal \(G_{\text{best}}\) in real time; the value of flag increases by 1 after every iteration;
Step 5: Judge whether it is precocious. If it is precocious, carry out precocious treatment, and turn to step 4, otherwise carry out the next step;

Step 6: Determine whether the maximum number of iterations is reached, if so, execute step 9, otherwise proceed to the next step;

Step 7: Judge whether it converges, if so, carry out the next step, otherwise, go to step 4;

Step 8: Determine the information of each particle corresponding to G_best;

Step 9: Place the virtual machine according to the value of G_best;

Step 10: Map the physical machine performing the task by the resource registration information;

Step 11: Physical machine executes scheduled tasks, ends tasks and reclaims resources;

Step 12: Output the result and end.

The overall implementation process is shown in Figure 3.

Figure 3. Improved particle swarm optimization implementation process in the task scheduling

5. Conclusion

This paper analyzes the characteristics and objectives of cloud platform task scheduling, starting with the task scheduling algorithm, proposes an artificial intelligence method for power dispatching automation system based on improved particle swarm algorithm, to build a cloud platform task scheduling optimization model. The task scheduling strategy based on the algorithm and physical model
fully considers the QoS requirements of cloud platform users and load balancing of cloud computing environment, which can effectively improve the efficiency of cloud platform task scheduling for of power dispatching automation system.

References
[1] YAN Ge, YU Jiong, YANG Xingyao. Two-step task scheduling strategy for scientific workflow on cloud computing platform[J]. Journal of Computer Applications, 2013, 33(4):1006-1000.
[2] Hazra D, Roy A, Midya S, et al. Distributed Task Scheduling in Cloud Platform: A Survey[J]. 2018, 20(4):83-102.
[3] Gao Y, Ma H, Zhang H, et al. Concurrency Optimized Task Scheduling for Workflows in Cloud.[J]. 2013, 65:77-89.
[4] Zhifeng Zhong, Kun Chen, Xiaojun Zhai, et al. Virtual Machine-Based Task Scheduling Algorithm in a Cloud Computing Environment[J]. Tsinghua Science & Technology, 021(006):660-667.
[5] Li J W, Qu C W. Cloud Computing Task Scheduling Based on Cultural Genetic Algorithm[J]. 2016, 35:28-48.
[6] Keshanchi B, Souri A, Navimipour N J. An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: formal verification, simulation, and statistical testing[J]. 2017, 124(February 2017):1-21.
[7] Liu Y, Xu X, Zhang L, et al. An Extensible Model for Multi-Task Service Composition and Scheduling in a Cloud Manufacturing System[J]. Journal of Computing & Information Science in Engineering, 2016, 16(4): 22-37.
[8] Phuoc Hung, Pham, Huh, Eui-Nam. An Adaptive Procedure for Task Scheduling Optimization in Mobile Cloud Computing[J]. Mathematical Problems in Engineering, 2015:1-13.
[9] Meifang Han. Application of Artificial Intelligence Detection System Based on Multi-sensor Data Fusion[J]. International Journal of Online Engineering, 2018, 14(6):31.
[10] Carlo Contaldi, Fatemeh Vafaee, Peter C. Nelson. Bayesian network hybrid learning using an elite-guided genetic algorithm[J]. Artificial Intelligence Review, 2018(293):1-28.
[11] Serena Yeung, N Lance Downing, Li Fei-Fei, et al. Bedside Computer Vision - Moving Artificial Intelligence from Driver Assistance to Patient Safety[J]. New England Journal of Medicine, 2018, 378(14):1271-1273.
[12] Seth D. Baum. Reconciliation between factions focused on near-term and long-term artificial intelligence[J]. Ai & Society, 2018, 33(4):565-572.