

Retraction

Retracted: Mobile Internet of Things Dynamic Grid QoS Service Matching Mechanism and Simulation Analysis

Complexity

Received 19 December 2023; Accepted 19 December 2023; Published 20 December 2023

Copyright © 2023 Complexity. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] X. Fan, "Mobile Internet of Things Dynamic Grid QoS Service Matching Mechanism and Simulation Analysis," *Complexity*, vol. 2021, Article ID 2884700, 11 pages, 2021.

Research Article

Mobile Internet of Things Dynamic Grid QoS Service Matching Mechanism and Simulation Analysis

Xuemei Fan 

School of Management, Jilin University, Changchun, Jilin 130022, China

Correspondence should be addressed to Xuemei Fan; fxm@jlu.edu.cn

Received 22 April 2021; Revised 18 May 2021; Accepted 4 June 2021; Published 11 June 2021

Academic Editor: Zhihan Lv

Copyright © 2021 Xuemei Fan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of the mobile Internet of Things, people have higher and higher requirements for services, and how to provide high-quality services that meet the needs of users has also become a hot issue of current research. This article is also in the context of the mobile Internet of Things, researching the solution to the service matching problem. In order to provide high-quality grid services, the grid system needs the support of quality of service (QoS). Based on the existing QoS matching strategy, a service matching strategy based on dynamic grid QoS is proposed. This paper combines semantic information, considers service quality QoS and user preferences, and improves the success rate of service matching through hierarchical matching, so as to provide users with high-quality services. At the same time, the classic Max-Min algorithm in grid task scheduling is analysed, and on this basis, the impact of QoS on task scheduling is fully considered, and combined with the concept of execution cost, a grid task based on QoS constraints (MOS) is proposed. Through simulation verification and analysis, the strategy in this paper not only achieved higher matching performance, but also improved the success rate of task execution, and the effect of system load balancing was also significantly improved.

1. Introduction

Internet of Things (IoT) is based on a large number of microdevices (such as actuators and sensors), combining traditional and emerging industry applications, with a new generation of Internet as the backbone, and is a ubiquitous and global network [1–3]. The emergence of the IoT provides a good basic platform for the realization of a pervasive computing environment. Therefore, the IoT is another wave of technology and economy brought by the global information industry after computers, mobile communication networks, and the Internet. It has brought great opportunities and challenges to the modern and fast-developing information digital life [4, 5]. Mobile IoT provides a good platform for ubiquitous computing. The purpose of pervasive computing is to establish an environment full of perception, communication, and computing capabilities, and gradually integrate this environment with people's lives [6]. An important direction in pervasive computing-service discovery [7] has become an important aspect of the mobile

IoT. For service discovery, the key is the service matching method, so how to perform service matching and how to provide users with high-quality services have become the hotspot and the most important research topic.

Since there are multiple grid resources with similar service functions in the grid system, the QoS of resources has gradually become one of the important factors that users need to refer to when selecting services [8]. Grid service is a service registered in the grid system by the resource provider in the description form of service or resource binding. Service matching based on QoS is the focus of research in resource management. Currently, there is no unified standard for the description, parameter definition, and service evaluation of grid services. Literature [9] studied the description of QoS parameters between service requesters and providers and used the method of sequential comparison for service matching. However, the matching process of this method is complicated and it is difficult to match high-quality services. Literature [10] analysed the QoS attributes of the service, combined the representative attributes in the

service into a matrix form, and obtained high-quality services through a local optimal selection algorithm. This algorithm has a faster matching speed, but the shortcomings are obtained. The service cannot be guaranteed to be optimal. Literature [11] uses a global optimization algorithm to obtain services with better service quality on the basis of considering global QoS factors, but the algorithm is slower and runs heavy.

On the basis of studying the existing QoS-based matching strategy, this paper proposes a service matching strategy based on dynamic grid QoS. This article combines semantic information to improve the success rate of service matching through hierarchical matching. At the same time, a grid task group scheduling algorithm (MOS) based on QoS constraints is proposed. The simulation results show that the strategy in this paper not only achieves higher matching performance, but also improves the success rate of task execution, and the effect of system load balancing has also been significantly improved.

2. Related Knowledge

In recent years, there has been an upsurge in the research and application of the IoT technology at home and abroad. Its user end extends and extends to any object and between objects for information exchange and communication [12]. With the promotion of users, operators and equipment manufacturers, Internet of Things services [13] appear in large numbers, and users cannot quickly and completely find services that meet their personal needs from a large number of services. This is called service discovery [14]. When faced with a large number of similar services, that is, the result of service discovery, users cannot accurately find the most satisfactory service. This is called service matching.

Regarding the algorithm research on service matching, the most influential one is the elastic matching algorithm [15]. The algorithm mainly considers service matching based on functionality, that is, considering the input and output parameters of the matching service. It defines four matching levels, namely, complete matching, plug-in matching, inclusive matching, and matching failure. Since there are multiple grid resources with similar service functions in the grid system, the QoS of the resources has gradually become one of the important factors that users need to refer to when selecting services. The literature [16] states that the scheduling system in the grid environment must adopt a hierarchical structure, which requires multiple scheduling instances to coordinate with each other, and analyses the properties of communication between scheduling instances. Literature [17] analysed the scheduling execution process that divided into three phases: resource discovery phase, system selection phase, and job execution phase. GGF [18] proposed a draft of the service grid scheduling architecture, explaining the relationship between OGSA-based scheduling services and other core grid. At the same time, the algorithm proposes that a complete grid scheduling system should include four functional modules: resource selection, scheduling queue generation, data management and

monitoring including I/O transmission and copy. The current resource organization models used in the grid computing environment are mainly as follows:

- (1) Hierarchical model: currently, the grid projects that adopt the hierarchical model include Globus [19], Legion [20], and AppLes [21].
- (2) Computational market/economy model: the grid computing market provides suitable tools and services to allow resource requesters and resource providers to express their needs. It can be used in conjunction with the hierarchical model to achieve more effective resource management.
- (3) Resource pool model: in this model, the information of all resources in the computing environment is recorded on a central server, and the description of the resources in the resource pool is disordered. Therefore, only work-level scheduling is supported in the condor environment, task-level parallelism is not supported, and communication between tasks is not supported.
- (4) EVP model of grid resource space: the EVP model proposed in Vega Grid includes three levels: effective resource layer, virtual resource layer, and physical resource layer.
- (5) OGSA resource service model: all resources exist in the form of services. Services are distributed in different environments and can be accessed through a universal resource identifier.

There are many current researches on service matching, and we can extend these methods to the IoT environment. So in this article, we will study the discovery, selection, and combination on the IoT platform in a semantic environment. The main idea of semantics [22] is to encode resources in a computer-process-able semantic form to expand the current human-readable information. Semantic service [23] is a server terminal in a client-server system that realizes machine-to-machine interaction through the World Wide Web. Semantic-based research will enable more advanced applications and functions. Computers will be able to search, process, and integrate the content of these resources more meaningfully.

Jini [24] is an object-oriented service discovery mechanism based on Java inheritance. It can change the hardware or software configuration of the service network that uses it at any time. This provides a perfect mechanism for the hardware or software that joins or exits the service network, and it also conveniently supports the configuration of the ubiquitous computing environment.

UPnP [25] is a standard promoted by Microsoft. As long as any device is connected to the network, other devices on the network will immediately know that there is a new device to join, and these devices can communicate with each other.

Salutation [26] is an open standard, and this standard has nothing to do with communication protocols, operating systems, and hardware platforms. Its purpose is to break the barriers between different technologies through this open

standard, so as to realize “universal computing” in the true sense.

Based on the above introduction to the existing service algorithms, it can be found that each has its own advantages and disadvantages in mechanism, and each has its own scope of application [27, 28]. Through analysis, it is concluded that they have the following shortcomings:

- (1) No semantic information added
- (2) Context-sensitive is not supported
- (3) Not suitable for public wide area network
- (4) Security and privacy

In order to solve the above problems, this paper combines semantic information, considers service quality QoS and user preference, and proposes a grid task group scheduling algorithm (MOS) based on QoS constraints. Through simulation verification and analysis, the strategy in this paper not only achieved higher matching performance, but also improved the success rate of task execution, and the effect of system load balancing was also significantly improved.

3. Dynamic Grid QoS Service Matching Algorithm

3.1. System Structure of Mobile Grid. With the gradual popularity of mobile Internet applications and the continuous improvement of mobile search applications, when users need to query some urgently needed, they will increasingly tend to use mobile search [29–34]. The structure of the mobile grid system is shown in Figure 1, which is mainly composed of three modules: a fixed grid site, a mobile terminal group, and a middleware mobile gateway.

In the mobile grid, mobile terminals have two purposes: one is as a resource to be scheduled; the other is as a resource for providing services. The former uses mobile devices to access grid resources, while the latter uses mobile devices as grid resources to provide services to other grid users. Mobile terminals refer to devices with limited computing resources, excluding notebooks. Mobile terminals continue to shrink in size, become more compact and easy to carry, but their functions are increasingly powerful, and their services continue to expand to integrate multiple functions such as communication, video, and reading. However, mobile terminals have shortcomings, which are manifested in the following points: limited memory and storage resources, usually between 2 and 64 Mbytes; computing resources, battery life, network bandwidth, display, and input capabilities are limited, and there are many kinds of processor and operating system.

3.2. Semantic-Based Resource Description Framework. The framework of this article provides grid services through interface definitions, which facilitates the access and combination of grid services. The framework determines the basic functions needed to realize sharing in a virtual organization, and the main relationships that exist between these different functions. However, the current framework

does not integrate semantic service technology, so there is a lack of semantic-based service discovery and sharing. Through the current research on grid technology and semantic grid projects, we have summarized the hierarchy of grid systems supporting semantic services under the existing OGSA, as shown in Figure 2.

The semantic-based resource description method can introduce clear formal semantics for resource description. By introducing semantics, it can enable grid users and grid applications to have a consistent common understanding of resources in the service grid environment. With the development of the mobile IoT, users will have higher and higher requirements for services. Personalized service matching methods will also have better applications. Therefore, considering the user’s preference in the service matching algorithm, it can also better reflect the human-centered thinking. On this basis, this paper introduces semantic information, considers QoS and user preferences, and adopts hierarchical matching to design a new service matching algorithm for mobile IoT. It is defined as follows:

- (1) Exact match: when the request concept is equal to the provided concept
- (2) Plug-in matching: when the requested concept is included in the provided concept
- (3) Containment matching: when the provided concept is included in the request concept
- (4) Matching failure: when it does not fall into the above three cases

Define the service S provided by a service provider as the following five-tuple:

$$S = \{S_t, S_i, S_o, S_Q, S_I\}. \quad (1)$$

Among them, S_t is the service type provided by the service provider, S_i is the input parameter of the service, S_o is the output parameter of the service, S_Q is the QoS parameter of the service, and S_I is the user’s preference for the service.

To define the service SRE requested by a service requester is also a five-tuple as follows:

$$SRE = \{SRE_t, SRE_i, SRE_o, SRE_Q, SRE_I\}. \quad (2)$$

Among them, SRE_t is the service type requested by the service requester, SRE_i is the input parameter of the service, SRE_o is the output parameter of the service, SRE_Q is the QoS parameter of the service, and SRE_I is the user’s preference for the service.

First, the matching of service types can filter out all unmatched services and provide accurate and similar matching services to the next step of matching, that is, matching the input and output of the service. The ontology theory is used here, and the matching of service types is carried out according to the matching of the semantic level. In this matching process, it depends on whether the service request and the service provided by the service provider refer to the same ontology. If not, the candidate service can be directly filtered out. If so, the function is directly applied:

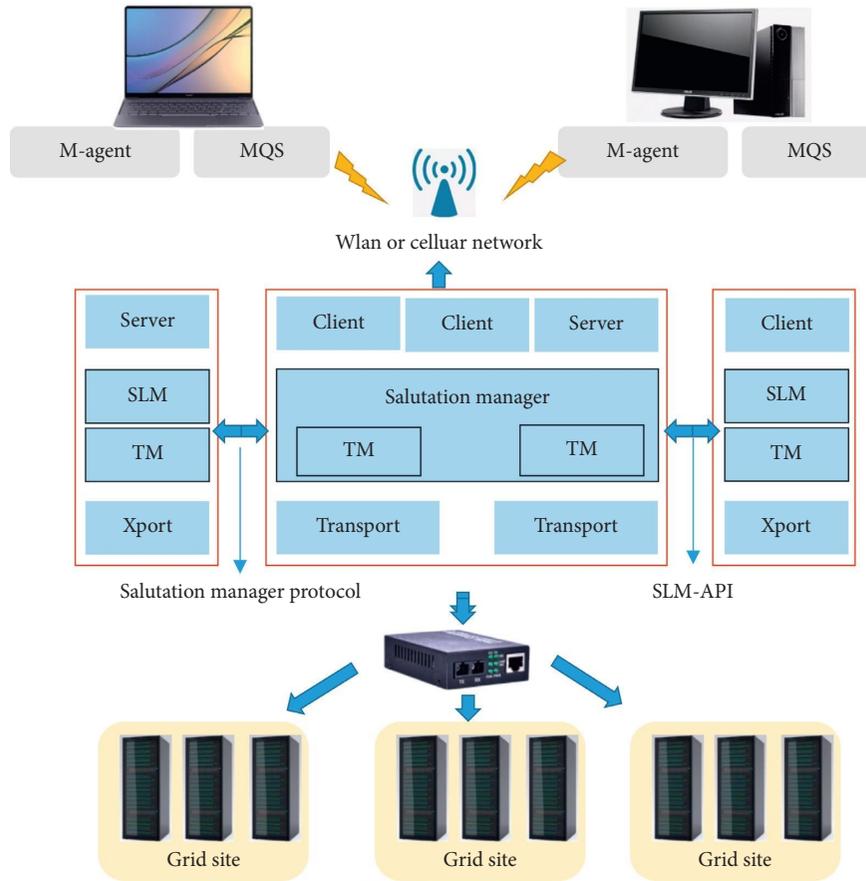


FIGURE 1: Structure diagram of mobile grid system.

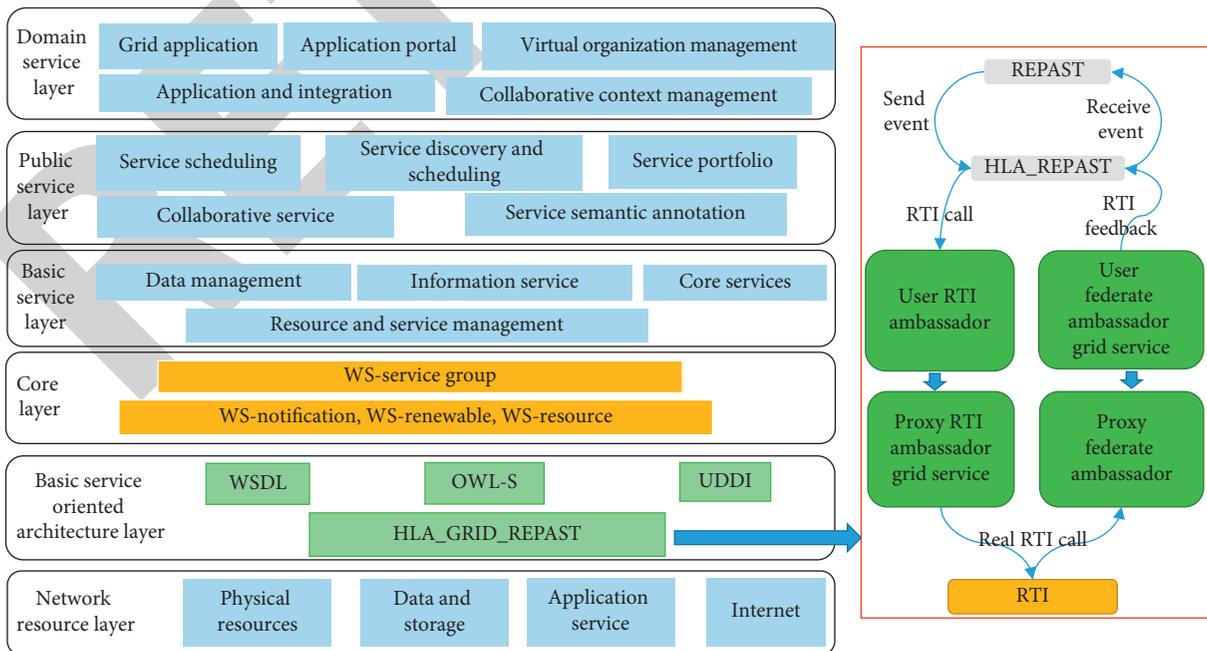


FIGURE 2: Hierarchical structure of OGSA system supporting semantic services.

$$\text{result} = \text{Match}(\text{SRE}_t, S_t). \quad (3)$$

If it is Fail, filter out the candidate service; otherwise, you can proceed to the next step of matching.

3.3. Service Input and Output Matching Degree Calculation.

After matching the types, the next step is to calculate the matching degree of service input and output, which is actually the problem of calculating the similarity between the two concepts in the ontology. Introduce the basic theory of ontology, so that semantic information is considered, matching at the semantic level is realized, and ambiguity is avoided. Suppose that, in an ontology concept tree, the parent concept directly contains the child concept. Suppose its root node is G , z_1 and z_2 are two child nodes, and there is a path $\{z_1, z_{1a}, z_{1b}, \dots, z_2\}$; then the similarity of concepts z_1 and z_2 is defined as

$$\text{similar} = \frac{\sum N_2}{\sum N_1}. \quad (4)$$

N_1 represents the number of subclasses of z_1 . N_2 represents the number of subcategories of z_1 . For multiple inputs and outputs, calculate the average value. In this way, the matching similarity of input and output can be calculated:

$$\text{result} = \text{Match}(\text{SRE}, S) = \frac{\text{similar}(\text{SRE}, S) + 1}{\text{similar}(\text{SRE}, S) - 1}. \quad (5)$$

In this way, the similarity matching degree of service input and output can be obtained through the above method.

3.4. QoS Parameter Matching. When there are multiple resource services that meet the requirements in the grid service system, in order to obtain higher-quality services, this article measures the QoS parameters globally from the aspects of grid service cost, execution time, and resource availability. User service preferences are increased to improve user satisfaction with services. Grid service quality is a measure of user satisfaction with the service.

Combined with the related requirements of QoS in the grid environment, the following 7 parameters are used to define and describe QoS:

- (1) c represents the grid service fee.
- (2) t represents the execution time of the grid service.
- (3) s represents the available grid resources.
- (4) a indicates the availability of grid resources.
- (5) r represents the reliability of grid resources. The calculation is shown as follows:

$$r = \frac{1}{N + 1} \sum_{i=1}^{\infty} \text{suc}(S_i). \quad (6)$$

Among them, N represents the number of grid services executed and S_i is the i -th service called by the user. When the execution is successful, the return

value of $\text{suc}(S_i)$ is 1; when the operation fails, the return value of $\text{suc}(S_i)$ is 0.

- (6) sr indicates the demand for various types of resources by grid services.
- (7) p is the user's preference for service parameters, expressed as a set as $P = \{p_1, p_2, \dots, p_n\}$, where n is the number of grid QoS parameters; the larger the value of p_i is set, the more the user prefers the parameter, and the better the parameter, the better it can meet the user's requirements.

Grid service quality parameter matrix: this refers to a matrix M composed of n grid services and m QoS parameters for each grid service. The representation of M is shown as follows:

$$M_{m \times n} = \{m_{11}, \dots, m_{mm}\}. \quad (7)$$

Among them, m_{ij} is a value between $[0, 1]$; a larger value indicates that the parameter of the grid service is better than the parameters of other services.

3.5. Matching of User Preferences. Grid services also have the characteristics of diversity, and different service matching strategies need to be adopted for different services, because users have different preferences for grid QoS parameters, that is, $P = \{p_1, p_2, \dots, p_n\}$ is different. Different users may have different preferences p_i for the same QoS parameters. Multiply the matrix M and the preference parameter column vector to obtain the preference-based global grid QoS parameter column vector M' . The representation of M' is shown as follows:

$$M' = P \times M^T. \quad (8)$$

3.6. Comprehensive Matching of Services. In this way, by combining the matching of the various stages above, a comprehensive matching of the final service can be obtained. The comprehensive matching degree of the service is calculated as follows:

$$\text{result} = \frac{a \times \text{Match}(\text{SRE}_t, S_t) + b \times \text{Match}(\text{SRE}, S)}{(1 - a - b) \times (\sum N_2 / \sum N_1)}, \quad (9)$$

where a and b represent weight information.

In this way, the comprehensive matching degree of the service is obtained, through which it can be judged which service is more suitable, so as to provide the service with the best service matching degree to the user. In actual matching, the requested service is hierarchically matched with all published services, and only those services that pass the previous level of matching can enter the next level of matching, thereby gradually narrowing the service set, and ultimately satisfying user requests. The service or service collection is fed back to the user.

3.7. Task Scheduling of Mobile Grid. In a grid computing environment, there are many independent workstations

distributed in different areas, and each workstation is composed of various hosts with different functions. These workstations from different regions are linked to each other through the WAN, sharing the services provided by each other. In order to improve the use efficiency of grid resources and quickly allocate the tasks to be completed to the appropriate resources, we need a suitable task scheduling algorithm. We propose an improved Max-Min algorithm for the grid task scheduling algorithm, and pursue the task scheduling model Makespan Oriented Scheduling (MOS) with the shortest overall task completion time. The algorithm execution process is as follows:

Step 1: according to the matrix $M'(t_i, r_j)$, the tasks to be allocated are sorted in descending order.

Step 2: sort and renumber the resources according to the execution cost of the resource host from small to large.

Step 3: according to different types of QoS requirements, classify T and put it into different queues. Assume that $Q=2$ means the highest demand, $Q=1$ means the second, and $Q=0$ means the lowest demand.

Step 4: starting from queue Q1, select t_i with the largest amount of tasks to start scheduling. Choose the resource r_j that can meet t_i 's QoS requirements and can be completed the fastest. If there are several units with the same completion time, choose the cheapest host.

Step 5: delete the task t_i from the non-empty queue Q1, and update the matrix at the same time; that is, update the matrix of the remaining tasks waiting to be scheduled on the r_j host.

Step 6: repeat the above steps to process the tasks in queues Q2 and Q3, respectively, until all tasks in all queues are scheduled, and the program ends.

4. Results and Discussion

4.1. User's Preferred QoS Service Matching. This section verifies the effectiveness of the above algorithm through experiments. For the QoS service matching process that does not include user preferences, the algorithm in this paper and the calculation methods proposed in literature [11], literature [12], and literature [15] are implemented. The comparison result is shown in Figure 3. It can be seen from Figure 3 that the preference-based QoS matching algorithm, on the premise of more satisfying the demand, takes longer than the matching execution time without adding user preferences; that is, the matching execution efficiency is higher.

4.2. Introducing Semantic QoS Service Matching. The matching algorithm uses a value between $[0, 1]$ to represent the result of each match; as long as it is greater than or equal to the threshold specified by the user, it can be output to the user. The algorithm does not only filter from the service type. It not only considers the matching of the input and output parameters of the service, but also considers the matching of

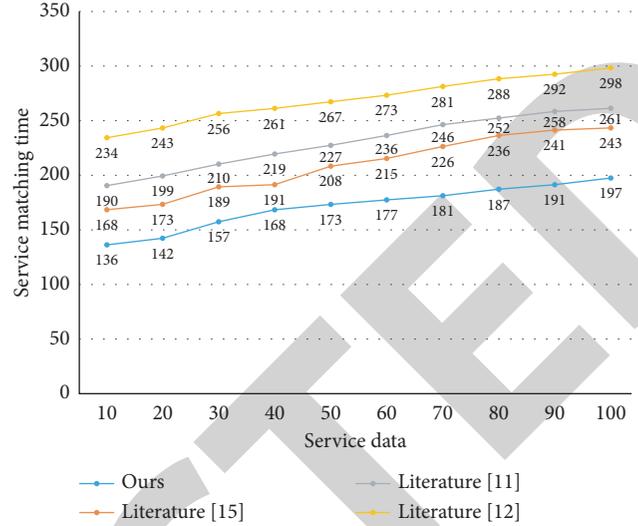


FIGURE 3: Grid service matching time comparison chart.

QoS, and more importantly, considers the user's preference. In this way, the found service can not only satisfy the user's request, but also be close to the user's preferences, thereby more satisfying the user's needs.

Based on the above aspects, the algorithm has an overall improvement in the accuracy of service discovery. After the introduction of semantic ontology, compared with grammar-based service discovery, it can match input and output while also matching related entities. By setting different matching thresholds, the scope of service selection can be scaled. From this point of view, the algorithm increases the range of services that may participate in matching, which also improves the recall rate of services. Figures 4 and 5 are comparison charts of the precision and recall of the algorithm in this paper and the comparison matching algorithm. It can be seen that the performance of the algorithm in this paper is the best.

4.3. Task Scheduling of Grid Resources. The scheduling process of the MOS algorithm is shown in Figure 6. Compare the traditional Max-Min scheduling algorithm and the MOS scheduling algorithm based on QoS constraints, as shown in Figure 7. It is not difficult to see that the overall completion time of the MOS algorithm has been greatly shortened. Compared with Max-Min algorithm, MOS scheduling algorithm also has a great reduction in execution cost.

Assume that the number of resources in the same grid domain is selected as 350, and the number of tasks increases from 50 to 500 in steps of 50. According to the above three grouping strategies, after grouping the number of tasks and when they are not grouping, the Min-Min algorithm and the algorithm in this paper are used to simulate, respectively, and the task execution success rate is compared, and the task execution success rate of 4 cases is obtained, as shown in Figure 8.

From the analysis in Figure 8, it can be concluded that as the number of tasks increases, the success rate of the two

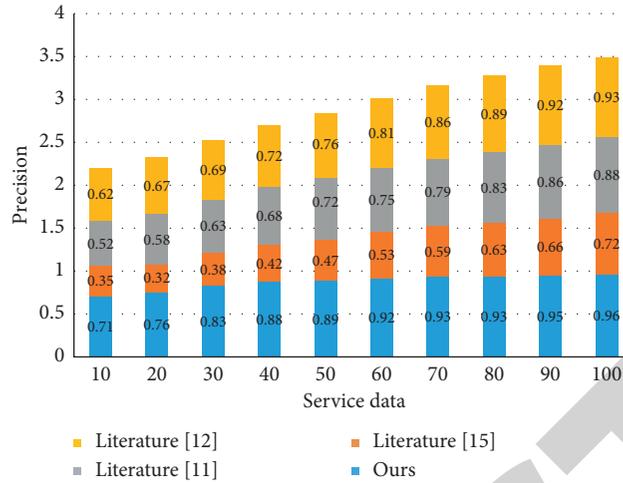


FIGURE 4: Comparison of recall rates.

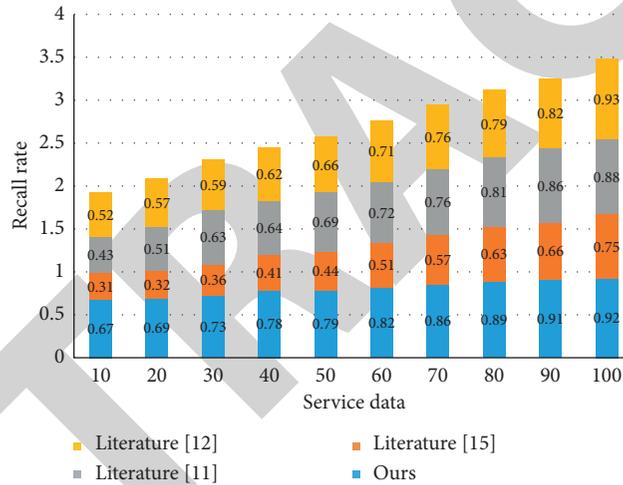


FIGURE 5: Accuracy comparison chart.

algorithms decreases. As the task is successfully executed, the energy of mobile resources is consumed. Because the energy of mobile resources is limited, it is unable to undertake too large tasks, so the more tasks, the lower the success rate of execution.

4.4. Performance Comparison with Other Matching Algorithms. In order to evaluate the performance of the dynamic grid QoS service matching algorithm proposed in this paper, a series of experiments are carried out to compare the method proposed in this paper with other methods. Randomly generate 10,000 similar services for each node, and randomly generate QoS attribute values for each service, and convert them into values within a certain range according to the standardization of the attribute values, which is convenient for calculation. In order to improve the accuracy, each experiment was run 500 times, and the most suitable service was selected from 10,000 services each time,

compare this method with Improved Genetic Algorithm (IGA) [35] and Artificial Neural Network Algorithm (ANNA) [36].

We conduct experiments on the basis that each node has 500 candidate services. Here, user feedback evaluation after use is a standard for us to measure the reliability of the algorithm. After each experiment, the user satisfaction of the service will be recalculated. As the number of trials increases, the user's satisfaction with the service will become more and more obvious; that is, the reliability of the algorithm will become higher and higher, and eventually become stable. As shown in Figure 9(a), we can clearly see that, with the increase of users, the performance advantage of this algorithm will become more and more obvious. To study a single node in the simulation combination, gradually increase the number of candidate services, and the corresponding time consumption will also increase. From Figure 9(b), we can see that the algorithm proposed in this paper is faster than the other two algorithms that have been significantly improved,



FIGURE 6: The scheduling process of the MOS algorithm. (a) Q1 queue scheduling situation. (b) Q2 queue scheduling situation. (c) Q3 queue scheduling situation. (d) Q4 queue scheduling situation.

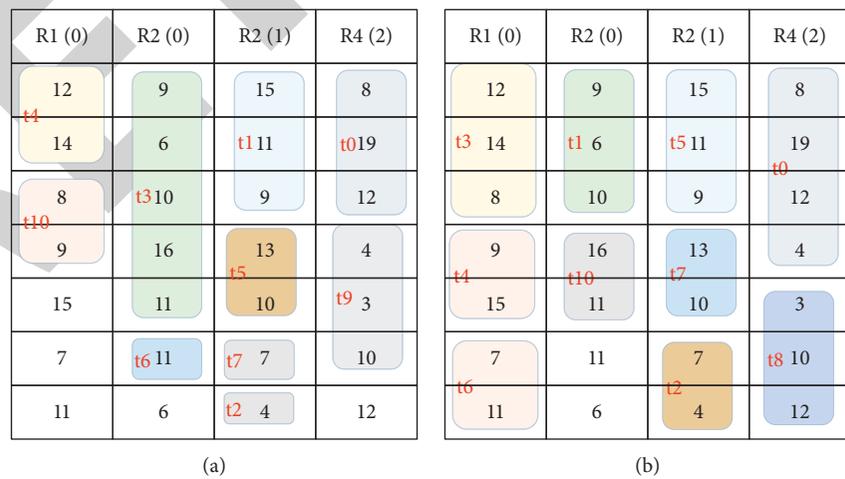


FIGURE 7: Comparison of scheduling between traditional algorithm and MOS algorithm. (a) Scheduling process of max-min. (b) Scheduling process of MOS.

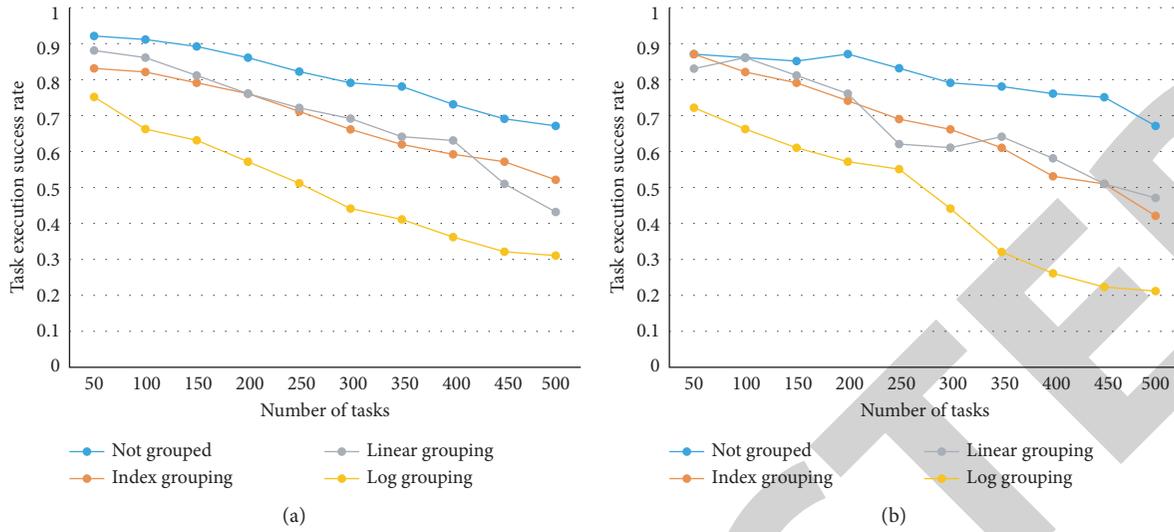


FIGURE 8: Comparison of the success rate of task grouping strategies under different task numbers. (a) Max-min. (b) MOS.

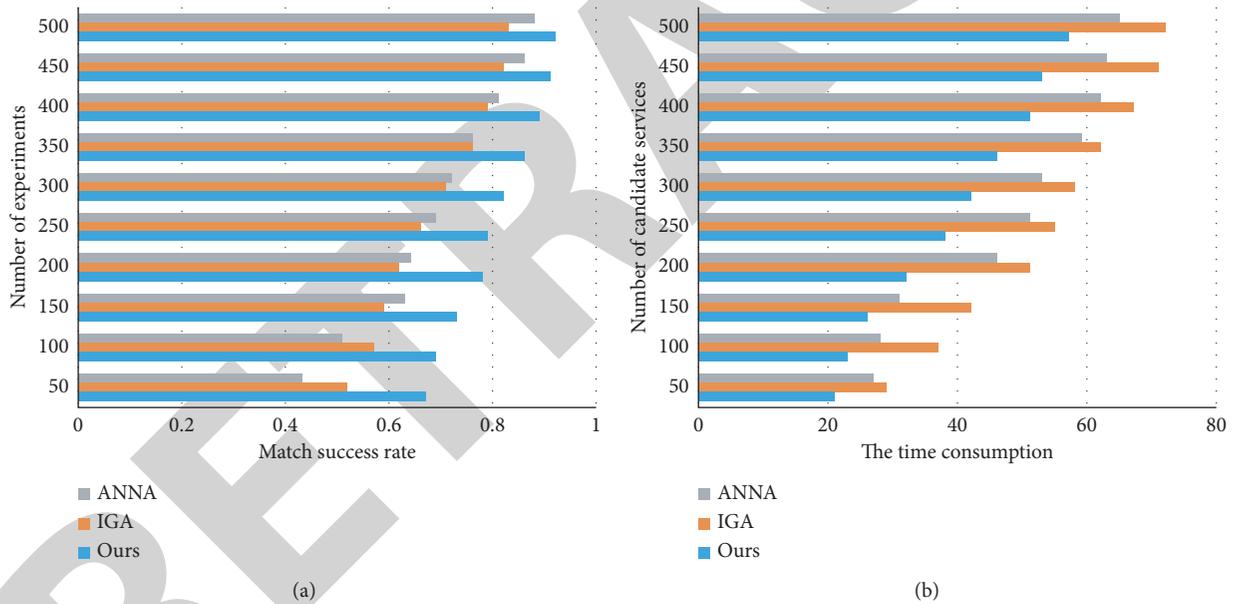


FIGURE 9: Comparison of algorithm reliability and matching time of a single optimal service. (a) Comparison of algorithm reliability. (b) Comparison of the selection time of a single optimal service.

enabling users to more quickly and accurately find the optimal service that meets their needs and achieve optimal matching.

5. Conclusion

Based on the analysis of existing QoS-based matching strategies, a dynamic grid QoS-based service matching algorithm is proposed. This article first combines semantic information, and not only considers the quality of service (QoS), but also considers user preferences, at the same time, fully considering the impact of QoS on task scheduling. A

grid task scheduling algorithm (MOS) based on QoS constraints is proposed. Through simulation verification and analysis, the strategy in this paper can meet the needs of users with a higher success rate and shorter scheduling time and achieve precise service matching. At the same time, the task execution success rate is improved, and the system load balancing effect is also significantly improved.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this study.

References

- [1] Y. A. Qadri, A. Nauman, Y. B. Zikria, A. V. Vasilakos, and S. W. Kim, "The future of healthcare internet of things: a survey of emerging technologies," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1121–1167, 2020.
- [2] A. Nauman, Y. A. Qadri, M. Amjad, Y. B. Zikria, M. K. Afzal, and S. W. Kim, "Multimedia internet of things: a comprehensive survey," *IEEE Access*, vol. 8, pp. 8202–8250, 2020.
- [3] A. Villa-Henriksen, G. T. C. Edwards, L. A. Pesonen, O. Green, and C. A. G. Sørensen, "Internet of things in arable farming: implementation, applications, challenges and potential," *Biosystems Engineering*, vol. 191, pp. 60–84, 2020.
- [4] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.
- [5] C. Savaglio, M. Ganzha, M. Paprzycki, C. Bădică, M. Ivanović, and G. Fortino, "Agent-based internet of things: state-of-the-art and research challenges," *Future Generation Computer Systems*, vol. 102, pp. 1038–1053, 2020.
- [6] S. Smys, A. Basar, and H. Wang, "Hybrid intrusion detection system for internet of things (IoT)," *Journal of ISMAC*, vol. 2, no. 4, pp. 190–199, 2020.
- [7] B. Pourghhebleh, V. Hayyolalam, and A. Aghaei Anvigh, "Service discovery in the internet of things: review of current trends and research challenges," *Wireless Networks*, vol. 26, no. 7, pp. 5371–5391, 2020.
- [8] Y. Choi, M. Choi, M. Oh, and S. Kim, "Service robots in hotels: understanding the service quality perceptions of human-robot interaction," *Journal of Hospitality Marketing & Management*, vol. 29, no. 6, pp. 613–635, 2020.
- [9] A. N. Abu-Safe and S. E. Elrofai, "QOS-aware meta-heuristic services selection algorithm and likert scale measurement for IOT environment," *International Journal of Computer Science Trends and Technology*, vol. 8, no. 1, pp. 1–8, 2020.
- [10] M. Singh, R. Kumar, and I. Chana, "A forefront to machine translation technology: deployment on the cloud as a service to enhance QoS parameters," *Soft Computing*, vol. 24, no. 21, pp. 16057–16079, 2020.
- [11] Z. Jia, S. Huang, and Y. Fan, "The ecological model and global optimization algorithm for service internet," in *Proceedings of 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pp. 1169–1174, IEEE, Hong Kong, China, August 2020.
- [12] S. Deng, Z. Xiang, P. Zhao et al., "Dynamical resource allocation in edge for trustable internet-of-things systems: a reinforcement learning method," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6103–6113, 2020.
- [13] T. Wang, L. Qiu, A. K. Sangaiah, A. Liu, M. Z. A. Bhuiyan, and Y. Ma, "Edge-computing-based trustworthy data collection model in the internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4218–4227, 2020.
- [14] C. Gong, F. Lin, X. Gong, and Y. Lu, "Intelligent cooperative edge computing in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9372–9382, 2020.
- [15] Y. Zhang, Z. Wan, X. Jiang, and X. Mei, "Automatic stitching for hyperspectral images using robust feature matching and elastic warp," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 3145–3154, 2020.
- [16] H. Wang, L. P. Zhou, Y. Zheng et al., "Double-layered supramolecular prisms self-assembled by geometrically non-equivalent tetratopic subunits," *Angewandte Chemie*, vol. 133, no. 3, pp. 1318–1325, 2021.
- [17] A. Samanta and J. Tang, "Dyme: dynamic microservice scheduling in edge computing enabled IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6164–6174, 2020.
- [18] P. Deane and M. Brinkerink, "Connecting the continents—a global power grid," *IEEE Power and Energy Magazine*, vol. 18, no. 2, pp. 121–127, 2020.
- [19] B. Crompe, A. Aristieta, A. Leblois et al., "Publisher correction: the globus pallidus orchestrates abnormal network dynamics in a model of Parkinsonism," *Nature Communications*, vol. 11, no. 1, pp. 1–14, 2020.
- [20] M. J. Taylor, "The evolution of the manipular legion in the early Republic," *Historia*, vol. 69, no. 1, pp. 38–56, 2020.
- [21] R. Rajão, B. Soares-Filho, F. Nunes et al., "The rotten apples of Brazil's agribusiness," *Science*, vol. 369, no. 6501, pp. 246–248, 2020.
- [22] Z. Zhang, Y. Wu, H. Zhao et al., "Semantics-aware BERT for language understanding," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 5, pp. 9628–9635, 2020.
- [23] A. Lober, H. Baumgärtel, and R. Verbeet, "Semantic service discovery in heterogeneous cyber-physical systems," in *Proceedings of Hamburg International Conference of Logistics (HICL) 2020*, pp. 591–623, Hamburg, Germany, September 2020.
- [24] A. B. Sallow, "Design and implementation distributed system using java-RMI middleware," *Academic Journal of Nawroz University*, vol. 9, no. 1, pp. 113–120, 2020.
- [25] G. Kayas, M. Hossain, J. Payton et al., "An overview of UPnP-based IoT security: threats, vulnerabilities, and prospective solutions," in *Proceedings of 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0452–0460, IEEE, Vancouver, Canada, November 2020.
- [26] D. Hillman, "Salutation and salvation in early modern theology," *Renaissance Quarterly*, vol. 73, no. 3, pp. 821–865, 2020.
- [27] J. Wen, J. Yang, B. Jiang et al., "Big data driven marine environment information forecasting: a time series prediction network," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 1, p. 1, 2020.
- [28] X. A. Hai, E. Hi, F. Xz et al., "AMD-GAN: attention encoder and multi-branch structure based generative adversarial networks for fundus disease detection from scanning laser ophthalmoscopy images," *Neural Networks*, vol. 132, pp. 477–490, 2020.
- [29] B. Li, R. Liang, W. Zhou, H. Yin, H. Gao, and K. Cai, "LBS meets blockchain: an efficient method with security preserving trust in SAGIN," *IEEE Internet of Things Journal*, p. 1. In press, 2021.
- [30] K. Zhang, J. Zhang, X. Ma et al., "History matching of naturally fractured reservoirs using a deep sparse autoencoder," *SPE Journal*, vol. 2021, pp. 1–22, 2021.
- [31] W. Wang, N. Kumar, J. Chen et al., "Realizing the potential of the internet of things for smart tourism with 5G and AI," *IEEE Network*, vol. 34, no. 6, pp. 295–301, 2020.

- [32] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast URL phishing detector: a convolutional neural network approach," *Computer Networks*, vol. 178, Article ID 107275, 2020.
- [33] J. Zhang, C. Shen, H. Su, M. T. Arafin, and G. Qu, "Voltage over-scaling-based lightweight authentication for IoT security," *IEEE Transactions on Computers*, vol. 99, p. 1, 2021.
- [34] J. Yang, T. Liu, B. Jiang, W. Lu, and Q. Meng, "Panoramic video quality assessment based on non-local spherical CNN," *IEEE Transactions on Multimedia*, vol. 23, pp. 797–809, 2021.
- [35] G. E. Weiqing and C. Yanru, "Task-scheduling algorithm based on improved genetic algorithm in cloud computing environment," *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, vol. 14, no. 1, pp. 13–19, 2021.
- [36] E. Shi, Y. Shang, Y. Li et al., "A cumulative-risk assessment method based on an artificial neural network model for the water environment," *Environmental Science and Pollution Research*, pp. 1–10, 2021.