Research and Implementation of Importing Distributed Cluster Data into Cloud Platform Based on JMS

Junyuan Zheng[1(✉)], Hongtao Ni[1], Min Sun[1], and Xinjian Chen[2]

[1] Suzhou City University, Suzhou, China
zhengjunyuan@szcu.edu.cn
[2] Soochow University, Suzhou, China

Abstract. This project researches and implements the problem of concurrent data repeated submission during the integration of distributed cluster system to a cloud platform. In this paper, three different solutions are proposed. Through analysis and comparison, the JMS service provided by Weblogic, which has an automatic failover function, is selected as the final solution to prevent concurrent duplicate data submission. And this solution only takes little code change effort on the existing system. When different users select duplicate records for concurrent submission, the message queue function of the Weblogic JMS service is used to record the records to be submitted. At the same time, the duplicate records are removed to avoid the problem of concurrent duplicate submission. Finally, the records can be imported in to the corresponding cloud platform through WebService technology.

Keywords: Distributed Cluster · Weblogic JMS · Cloud Platform · Concurrent and Repeated Submission · WebService

1 Introduction

With the rapid growth of the system service volume, the traditional single-machine deployment mode is unable to meet business needs. At this time, distributed cluster deployment mode arises [1]. Multiple services are deployed on various servers, and each service is a node. This way, when N nodes are deployed, the business processing capacity is increased by N times. The collection of these nodes is called a cluster. The cluster environment is a good choice for improving the business processing capacity [2]. A PLM product in this project is a cluster system deployed on Weblogic. However, with the further development of Internet technology, more and more systems are migrating to the cloud. Compared with most prefabricated data centers, the cloud platform is cheaper, more stable, more secure, and more scalable. This project is to import the data in a PLM system into the Oracle fusion PD cloud platform. So this paper mainly introduces the asynchronous message processing technology based on JMS [3] to prevent different users in the distributed cluster system from selecting duplicate records for concurrent submission and import.

S. Yu et al. (Eds.): TridentCom 2022, LNICST 489, pp. 73–83, 2023.
https://doi.org/10.1007/978-3-031-33458-0_6

2 Core Terms

2.1 JMS

Middleware is a software technology between client and server [4], message middleware is a middleware technology composed of message transmission mechanism or message queue mode [5], and Java Message Service (JMS) is a specification proposed by Sun company to unify various information middleware interfaces, which provides a set of interfaces independent of specific implementation [6], which is used to connect two application programs, or send messages in a distributed system. JMS is a vendor-independent API that performs asynchronous communication to access, send, and receive system messages. Because the data transmission process is not synchronized [7], the message sender can send a message without waiting for a response. The message sender sends the message to a virtual channel (queue or topic), and the message receiver listens to or subscribes to the channel. The message may eventually be forwarded to one or more message receivers, who do not need to respond to the message sender [8]. The use of Weblogic JMS can be divided into two parts. One part is that Weblogic publishes the corresponding JMS service as a server to provide services for the client program. The other part is that the application publishes messages as a client to the JMS server or obtains statements from the JMS server for consumption [9].

2.2 WebService

WebService is the interaction between network applications based on the HTTP protocol through the Internet. It uses WSDL (Web service definition language) and soap (simple request protocol) to realize the call between different languages [10]. WebService is a network-based, distributed, and modular component. It performs specific tasks and complies with detailed technical specifications. These specifications enable WebService to interact with other compatible components. It mainly uses HTTP and soap protocols to transmit business data on the web. Soap invokes business objects through HTTP to perform remote function calls; Web users can use soap and HTTP to call remote objects through methods called by the Web [11].

3 Method

3.1 Scheme Demonstration and Selection

This project realizes the system integration function of a PLM system to the fusion cloud platform. This PLM product has a DFCO object that needs to be imported into the item object in the fusion cloud system. The project schematic and other files uploaded by the user under the publish workspace tab on the DFCO object, such as .DRW files and .PRT files, users can search and select an item number in the fusion cloud platform system as the corresponding item object to be imported. After clicking publish, the complementary relationship between the two can be imported through WebService, and the files will also be imported synchronously. When importing objects with attachments such as engineering schematics, due to the size of the attachment capacity, the import

takes a specific time (usually a few seconds). At first, it is set to import only a single piece of data at a time. Because the import time of a single record is short, the problem of repeated submission can be ignored. The customer feels that the efficiency is too low in the process of using, so the system must select several records to import simultaneously. However, if multiple records are submitted together, the submission processing time will be longer, and it is easy for multiple users to submit duplicate records concurrently. That is, user a is submitting 1 and 2 records in DFCO1, but the submission has not been completed. User B also selects duplicate records in DFCO1 for submission because there are duplicate data in them, which will lead to the problem of data duplication and confusion.

According to the needs of customers, three solutions were studied and designed during the development of this project. The solutions and comparison are listed as below:

Scheme 1: Creating a static array list. Because the static array list is stored in the method area of the Java virtual machine, the elements contained in the method area are always unique in the whole program and are shared by all threads; it is very appropriate to judge duplicate records [12]. All DFCO ID numbers submitted in this project can be put into this static array list before submission. When another user submits, the selected ID number will be compared with this static array list. If there are no duplicate records, the ID number submitted by the user will also be added to this static array list. If one or more identical ID numbers exist in the static array list, a warning message will pop up to tell the user that duplicate records are in the submitted data. Duplicate records will not be submitted, but only non-duplicate records will be submitted. After completing the submission, delete the corresponding ID number from the static array list. This solution has no problem for single-machine systems. However, the PLM system of this project is a distributed cluster system deployed on Weblogic. The distributed cluster system is composed of one admin node and one or more slave nodes, and the PLM system is deployed on each node, so there will be a static array list on each node; that is, there will be multiple static arrays lists exist at the same time, and the DFCO ID submitted on different nodes will only exist in the static array list on the corresponding node. Multiple static arrays lists cannot be compared to obtain duplicate records. Therefore, this solution can not play a role in comparison and deduplication for distributed cluster systems.

Scheme 2: The popular message queues rabbitmq, rocketmq, and ActiveMQ [13] are used to control the submitted records through message queues. This method checks whether the same ID number exists in the message queue before submitting records in batch. If it does not exist, the ID numbers of the records to be submitted are all stored in the message queue, and then the records are taken from the message queue one by one and synchronized to the fusion cloud System one by one. Suppose the record ID to be submitted and the ID in the message wise are found to have duplicate records through comparison. In that case, a warning message will pop up to tell the user that there are duplicate records in the submitted data. Duplicate records will not be submitted. Only non-duplicate records in the message queue will be submitted. Because there is only one message queue in the distributed system, this solution can meet the requirements of the distributed cluster system. However, to prevent the repeated submission of such a small function, the jar package of the corresponding message queue needs to be additionally introduced. At the same time, the service of a message queue needs to be deployed and

maintained. The big code changes need a lot of effort. Moreover, the import function is highly dependent on the message queue service; Once there is a problem with the message queue service, the import function cannot be used. The coupling is too strong, and no backup message queue is available.

Scheme 3: As this PLM product is deployed on Weblogic, Weblogic has very powerful functions as an application server [14]. It has its own JMS service. Users can configure the JMS service to realize the function of the message queue. The JMS function is similar to scheme two's working principle of the message queue and will not be described here. In this scheme, the code changes are minor. Meanwhile, Weblogic has the functions of redundant backup and automatic failover when the server where JMS is located (usually the admin node) goes down, it will automatically change one of the slave nodes into a new admin node. At the same time, move all services to the new admin node (including the configured JMS service). In this way, the system can be ensured to have high availability in the actual project application; It can also play the role of data disaster recovery backup.

Through the comparison of the above three schemes relying on static array list, third-party message queue, and Weblogic's built-in JMS service, it can be seen that scheme three can solve the problem of repeated data submission with minimal changes in the existing technology and successfully realize the import of cloud platform data. Therefore, this solution is selected for this project.

3.2 System Design and Implementation

The specific architecture block diagram of the project system is shown in Fig. 1.

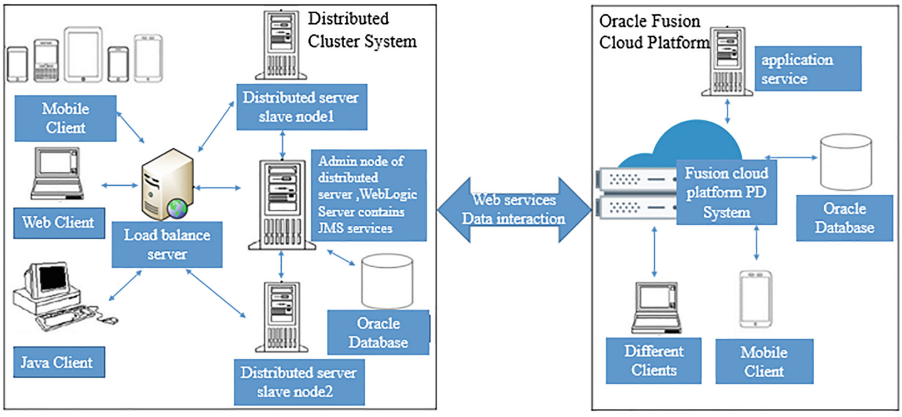

Fig. 1. System architecture block diagram.

The PLM distributed cluster system of the project is shown in the left block. It is composed of a load balance server, admin node, and multiple slave nodes, which are deployed on the WebLogic Server [15], and the JMS service is deployed on the admin node. The right block is the Oracle fusion PD cloud platform system. This distributed

cluster system integrates data through WebService to import data to Oracle fusion cloud platform.

The above system architecture diagram shows that the JMS service is only deployed on the admin node, it is unique in the distributed cluster system. It can be used to record the submitted data and perform deduplication. When different users perform the publish action, that is, data migration, the ID corresponding to the selected data in the database will be put into the JMS message queue for queuing. Because the ID of each data in the database is unique, each time the newly submitted data ID is put into the message queue, it will be compared with the existing data ID in the JMS message queue. If there is a duplicate data ID, the duplicate data ID will be discarded, and only the non-duplicate data ID will be added to the JMS message queue for subsequent migration. After completing the data migration, remove the corresponding data ID from the JMS message queue. The problem of concurrent duplicate submission is avoided by caching the JMS message queue, and then removing the duplicate data ID.

JMS Configuration Method

Parameter Allocation Method

Modifying the Weblogic configuration file on the server side of the corresponding distributed system can realize the role of deploying JMS services on Weblogic. For example, the configuration file path of the distributed server in this project is: $cluster-name/$agile_Home/agiledomain/config/config.xml. Add the following contents to the config.xml file:

```
<cluster>
   <name>$ClusterName</name>
  ...
</cluster>
<jms-server>
   <name>C4cJMSServer</name>
   <target>$ClusterName</target>
   <persistent-store>C4cStore</persistent-store>
</jms-server>
<jdbc-store>
    <name>C4cStore</name>
   <prefix-name>C4C_</prefix-name>
   <data-source>AgileHaDataSource</data-source>
   <distribution-policy>Distributed</distribution-policy>
   <target>$ClusterName</target>
   <migration-policy>Always</migration-policy>
</jdbc-store>
```

After restarting the server, you can see the JMS server on the corresponding Weblogic console. From the configuration content added above, you can see that the JMS server configured in this project is named C4cJMSServer, as shown in Fig. 2.

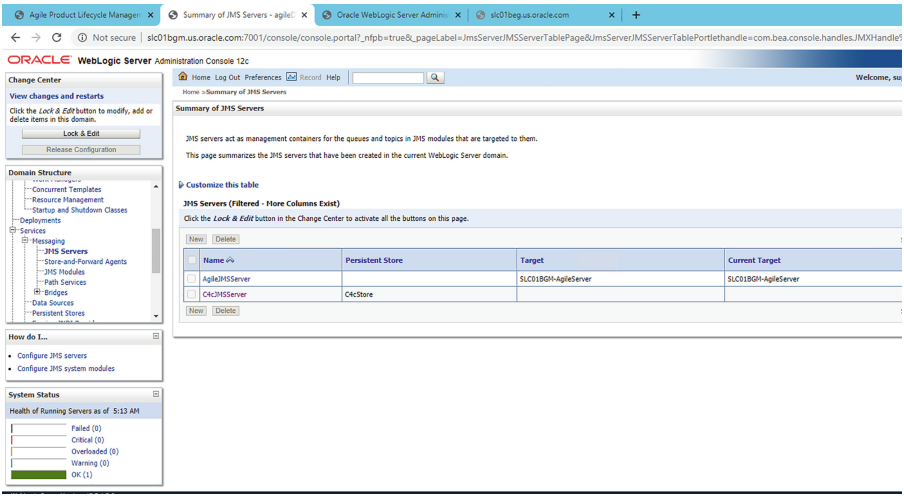

Fig. 2. JMS Server.

Console Configuration Method

In addition to the above method of modifying the configuration file, you can also configure the JMS server through the interface on the Weblogic console. Find the corresponding domain under the domain structure on the left of the Weblogic console, enter the service, and then enter the JMS server under the messaging. Then click the "new" button in the list on the right to enter the navigation page of the new JMS server, and then follow the prompts step by step [16]. The manual configuration method will eventually be written into the configuration file. The effect is the same as the method of modifying the configuration file parameters mentioned in Sect. 2.2.1.1. After the configuration, you can also see the configured JMS service, such as C4cJMSServer on the console.

Cloud Platform Data Import

Weblogic deploys the framework of Java-based WebService. The distributed system and cloud platform exchange data through WebService. The WSDL file of this standard-based interface modular WebService is easy to manage and maintain, and it can ensure compatibility between different clients (.Net, Java, and BPEL). In addition, the batch API access mechanism can provide better access performance.

Before data migration, the user can map the data to be migrated to the item object of the cloud platform by modifying the parameters of the configuration properties file on the distributed system side. For example, a parameter can be defined CLOUD_ CAD_ INTEGRATION_ ENABLED, the value of this parameter can be set to be true or false to enable or disable this import function, and through the parameter CLOUD_ INTEGRATION_ APPLICATION_ APINAME = OracleCloud and CLOUD_ PD_ ITEM_ REF_ OBJECT_ SUBCLASS_ APINAME = PDItem to control the API interface of the corresponding application. Define the WebService objects that can be queried, accessed, and operated in the WSDL file of the cloud platform. For example, the item object in the cloud platform in this project is shown in Fig. 3.

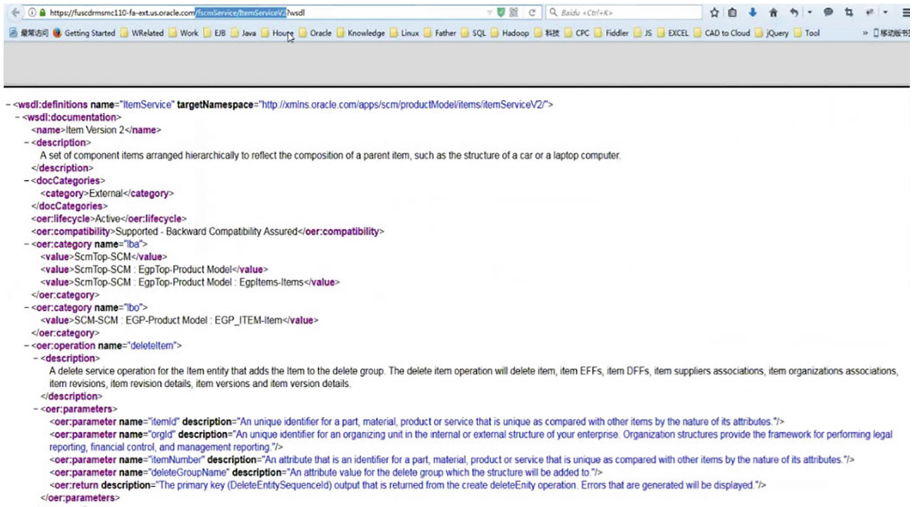

Fig. 3. WebService access item object configuration.

Then, build the jar file generated by compilation and put it into the distributed system server. At the same time, import the Keystore in the WSDL file of the cloud platform's WebService into the distributed system. Only in this way can the access be trusted. The WebService operations are mainly divided into 2 steps: (1) Search and query functions. The distributed cluster system accesses the cloud platform system, and the cloud platform query data in the distributed cluster system. (2) Data import. This step import data from the distributed cluster system to the cloud platform. The two detailed steps analysis are as follows:

(1) Search query function (cloud platform -> distributed cluster system)

On the distributed system side, you can query the item object of the cloud platform defined by the WSDL file to be operated. At the same time, you can map the attribute in the distributed system to be migrated with the attribute of the corresponding cloud platform's WebService, as shown in Table 1.

Table 1. Attribute relation correspondence table.

Reference object attribute	Webservice attribute
Name	ChangeNotice
Description	Description
Current Status	StatusTypeValue

When searching in the distributed system, the data search attribute returned from the cloud platform is the corresponding three attributes in Table 1.

(2) Data import (distributed cluster system - > cloud platform)

The WebService corresponding to the cloud platform is configured on the distributed system side, the operations and messages are described abstractly, and then the standardized XML is used for message transmission. The distributed system side data, such as .DRW and .PRT files, are imported into the attachment tab under the item object corresponding to the searched cloud platform in Sect. 3.2.2 (1). By clicking the Publish button on the DFCO object, the XML format data can be transmitted through HTTP protocol to realize the reading and writing operation of WebService. The whole publishing process is divided into three steps, namely, verification, association, and import. The function implementation flow chart of importing specific data from distributed cluster system to Oracle fusion PD cloud platform is shown in Fig. 4.

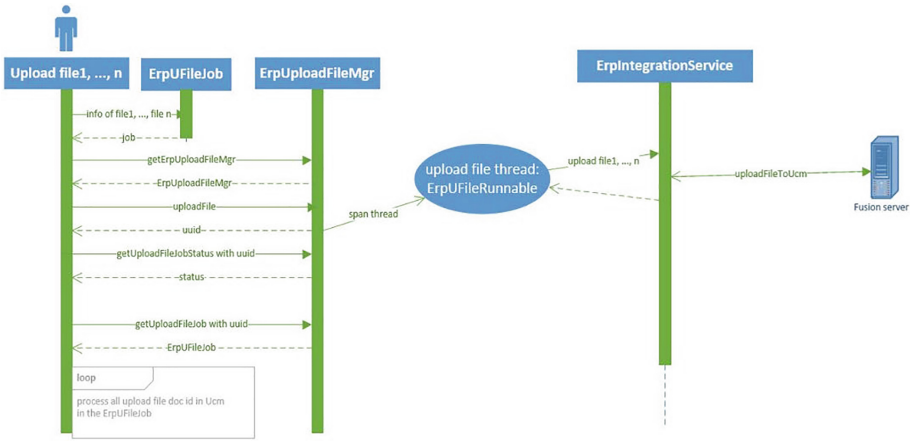

Fig. 4. Data import flow chart.

4 Results

After the system is implemented through the above scheme, creating a new DFCO00029 object at the distributed cluster system side, and adding the attachment files P00003.PRT and P00003.DRW under this object. Then, there are two attachment files P00003.PRT and P00003.DRW which designed by the CAD drawing software under the publish workspace tab of the DFCO00029 object in the distributed system. Configuring the JMS service and WebService with this paper's solution on the Weblogic. After that, the DFCO00029 object to associated Oracle fusion Cloud platform, we can see the results as shown in Fig. 5.

Analyzing Fig. 5, it can be seen the two attachment files P00003.PRT and P00003.DRW associate the "Item Number" field with the item object "pd_itemtest6

and pd_itentest110" of the fusion cloud platform through the WebService technology in Sect. 3.2.2 of this paper.

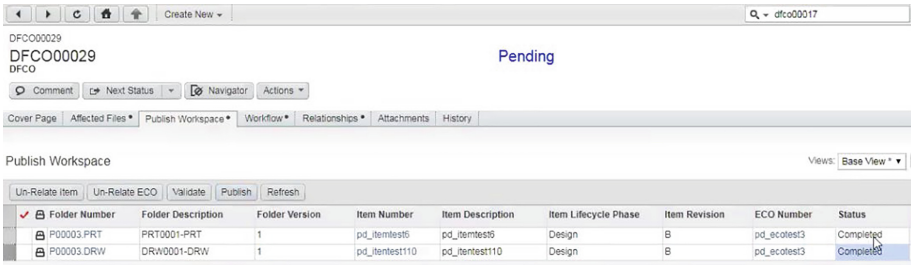

Fig. 5. Distributed cluster import interface.

At the same time, migrate the two records P00003.PRT and P00003.DRW to the oracle fusion cloud platform. If another user logs in to the distributed cluster system through other clients and submits the imported records (P00003.PRT and P00003.DRW) repeatedly, the warning message "selected row (s) is being published and cannot be added to a new job. Selected rows not being published will be added to a new job" will popup on the user interface as shown in Fig. 6, duplicate records will not send to the JMS message queue, which plays a role in preventing duplicate submission.

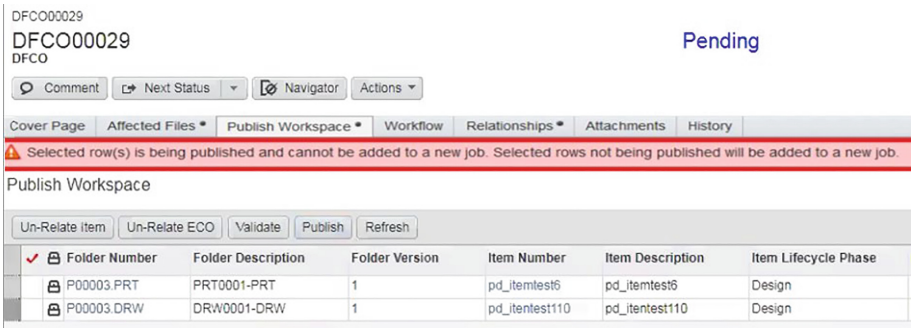

Fig. 6. Duplicate data submission warning.

After the distributed cluster system prompts the import completely, you can see all the item objects just imported under the "change order" object corresponding to the Oracle fusion cloud platform. The relationship has been successfully imported, as shown in Fig. 7.

The corresponding records (attachments P00003.PRT and P00003.DRW) successfully imported to the corresponding items in the cloud platform, the attachment record P00003.PRT is shown in the pd_itemtest6 of the Oracle fusion cloud platform, as shown in Fig. 8.

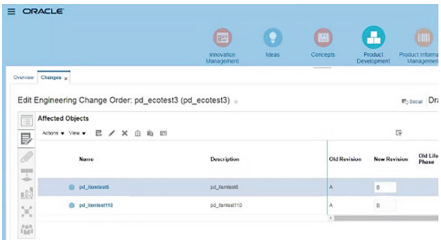

Fig. 7. Import the item object of the cloud platform.

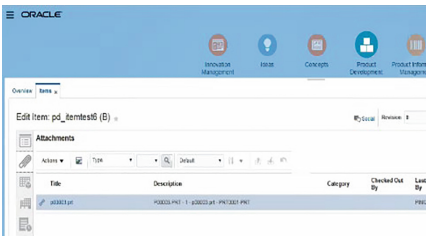

Fig. 8. Import file of item object.

5 Conclusions

This paper proposes a specific implementation method of importing distributed cluster data into the cloud platform based on Weblogic JMS and WebService technology. When importing data, the problem of duplicate submission of concurrent data is effectively prevented through Weblogic JMS. The scheme provided in this project is used to migrate the cloud platform of the actual project, which can effectively and accurately realize the data import function and prevent the problem of duplicate submission of concurrent records. Through this method, users can meet the actual application requirements of importing the data and information on the products of the distributed cluster system to the cloud platform, which not only saves the development cycle but also is easy to realize, thus significantly improving the production efficiency and quality of the software.

References

1. Wu, G.-F., Ding, J.-H., Xu, Y.-B.: Distributed data synchronization based on JMS. Comput. Syst. Appl. **24**(01), 171–175 (2015)
2. AN, B., MA, J., CAO, D., et al.: Towards efficient resource management in virtual clouds. In: 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops(ICDCSW) on Proceedings, pp. 320–323. IEEE, Atlanta (2017)
3. Luo, X.-B., Dong, S.-B., Xu, H., Zhang, L.: Asynchronous message processing technology and application based on JMS. Comput. Eng. 2002(12), 121–122+149 (2002)
4. Wang, L.-L., Xiao, C.-J., Zhang, X.-W.: A message-oriented secure transmission middleware model. Comput. Sci. **2007**(07), 288–292 (2007)
5. Li, Y.-M., Ling, J.: Design of security message middleware. Comput. Eng. Design **32**(06), 1934–1937 (2011)
6. Deng, Y.-K., Wang, C.: Research and design of JMS-based secure communication model. Comput. Eng. Design **30**(15), 3526–3530 (2009)
7. Shang, X.-N., Sun, L.-Y., Peng, T., Liu, C.: Design and research of a secure communication scheme for asynchronous network system. Comput. Eng. Sci. **34**(11), 34–37 (2012)
8. Zheng, L., Chen, Y.-L., Liu, R.-M., Zheng, D.-Y.: Research on cloud service of message queue based on open source. Inf. Commun. Technol. Policy **311**(05), 52–56 (2020)
9. Zheng, J.-Y.: Implementation of PLM product WebService access on oracle cloud platform. Comput. Knowl. Technol. **15**(07), 37–39 (2019)
10. Tan, Y.-J.: Research on java message communication. Comput. Program. Skills Mainten. 414(12), 40–41+61 (2019)

11. Peng, X.-T.: The design and realization of electronic document archiving interface based on XML and JMS. In: 3rd International Conference on Advanced Education Technology and Management Science (AETMS) on Proceedings, pp. 418–422. Science and Engineering Research Center, Beijing (2016)
12. Liu, J.-C., Xavier. R.: An array content static analysis based on non-contiguous partitions. Comput. Lang. Syst. Struct. 47, 104–129 (2017)
13. Zhang, J., Man, S.-G., Liu, K., Zhou, L.-J.: Using message queue to achieve data consistency method. Comput. Syst. Appl. **28**(09), 185–189 (2019)
14. Ni, H.-T: Deployment of Web project on ORACLE cloud platform. Wirel. Internet Technol. 15(06), 18–19 (2018)
15. Somogyi, A., Revanuru, N., Irudayaraj, R., Felts, S., Zhou, T., Zhao-Perez, F.: Patent issued for system and method for using a gridlink data source to connect an application server with a clustered database. US20120066363 A1, US (2015)
16. Elim168 blog page. https://blog.csdn.net/elim168/article/details/72674355. Accessed 21 July 2022
17. Su, Z.-F.: Research on heterogeneous database integration scheme based on Open URL protocol and XML Schema. Comput. Eng. Design **29**(16), 4308–4310 (2018)