The Distributed UDDI System Model Based on Service Oriented Architecture

Yongxin Feng
School of Information Science and Engineering
University of Shenyang Ligong
Shenyang,Liaoning Province, China
fengyongxin@263.net

Qin Li
School of Information Science and Engineering
University of Shenyang Ligong
Shenyang,Liaoning Province, China
liqin142231@163.com

Abstract—**Aiming at the issues of the traditional centralized UDDI model, the distributed UDDI model was proposed. The model adopted the latest version specification of UDDI v3 and the management technology of key space, avoiding uneven service accessing, low processing efficiency and prone to single point of failure. Moreover, service publication and service discovery were elevated, and the good practicability of distributed UDDI system can be verified.**

Keywords-Centralized UDDI; Distributed UDDI; Service Publication; Service Discovery

I. INTRODUCTION (HEADING 1)

SOA[1] (Service Oriented Architecture) enables various service components in an independent, loosely coupled way to integrate, and further achieve more powerful and complex business need so as to adapt to the heterogeneous, complex and changeable environment of various software systems. SOA mainly contains three essential roles, including service provider, service requester and the UDDI registry center [2,3]. The UDDI registry center is a vital link between service provider and service requester [4], which enhances the ability of parallel processing for large amounts of data, also ensures the accuracy of the data. UDDI is an essential component of SOA [5]. It provides users with a Web Service description, discovery and integration [6,7] through a unified programming interface. However, many problems still exist in the traditional model. First, a large number of network services register on a centralized service registry data center, the processing of service request can not balance access, then will lead the service registry performance bottlenecks. Second, when the concentrated service registry nodes fail, the registered information on registry center has not been backed up on other registry centers, resulting in a large number of registered information lost. As for the above limitations of the traditional centralized UDDI system model, a three-layer structure of distributed UDDI interoperable model is proposed, the distributed technology [8,9] which improves the design and the implementation of a distributed UDDI system model.

II. THE WEB SERVICE MODEL

The standard [10] implementation of the model for Web Services is that the service provider has an accessible via network software module (that is, the implementation of Web Service), which defines the service description and then publishes the description in the service registry. The service requester finds the service registry center to get the service description, uses the information to complete the binding with the service provider, achieves the Web Service interaction and calls the operations. In the Web Service system, all the applications are abstracted into services that include three roles and three kinds of operation. Figure 1 shows the architecture of Web Services.

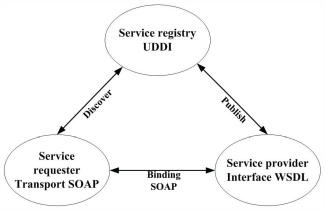

Fig.1 Architecture of Web Services

Service provider: in the perspective of commercial, service provider is a business entity in the development of services; in the perspective of architecture, service provider is a platform for managed access services or a service running environment. Service requester: in the perspective of architectural, this is an application to find and invoke services. Service registry center: service providers release their own described services on it.

Publish: in order to make the service available, the service providers need to publish service description so as to enable the service users can find it. Inquiry: in inquiry operation, service requester can get a service description directly or inquiry the required type of service in the service registry center. Binding: in the binding operation, the service requester use the binding details about the service description to locate, contact, and invoke the services, thereby invoking or starting the interaction with the service at runtime.

Currently, the implementation of registry systems is mostly based on a centralized design. It consists of a server and a client, and stores all the registered service information across the entire network on the server database. The centralized UDDI system architecture is illustrated in figure 2.

978-1-4673-9904-3/16/$31.00 ©2016 IEEE

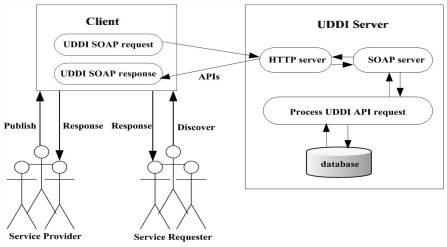

Fig.2 The design of centralized UDDI system architecture

In this system, the service provider and service requester are usually to implement service publication or service discovery operations in the UDDI Client, which needs to call the SOAP API provided by UDDI Server to handle the service publication or service discovery. The UDDI Server can be implemented in the six SOAP APIs [11,12], including API UDDI Inquiry, API UDDI Publication, API UDDI Security, API UDDI Custody Transfer, API UDDI Subscription and UDDI Subscription UDDI Replication. The most commonly used in these six kinds of API function are UDDI Security, UDDI Publication and UDDI Inquiry.

When service requester and service provider send service inquiry or service publication to the centralized UDDI system, usually five kinds of data structures including BusinessEntity, BusinessService, BindingTemplate, TModel and PublisherAssertion are used. These structures are the most critical factors in the UDDI specification, and all service information is defined by them. Figure 3 shows the information about the description and the relationship, basing on the above five structures.

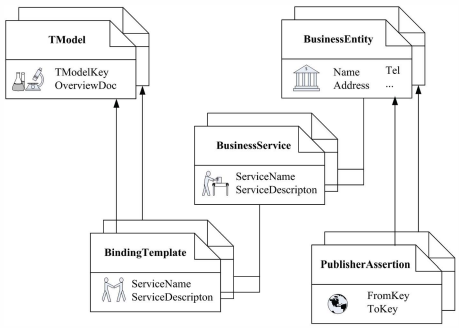

Fig.3 UDDI information model

BusinessEntity (the entity data structure) is the service provider's organization information mainly covers the organization name and other relevant information. BusinessEntity can publish multiple BusinessService (the service data structure) in UDDI system. BusinessService mainly includes the service name, the description information of the service and the classification information. TModel (the technical fingerprint data structure) mainly includes some information that is related to the classification. BindingTemplate service can achieve binding operation of the data structure and the technology fingerprint. BindingTemplate also describes the classification of data structure service information.

The relations among three data structures of BusinessEntity, BusinessService and BindingTemplate are related to each other. The structure of BindingTemplate needs to refer to the information of TModel structure, and PublisherAssertion structure needs to refer the information of BusinessEntity structure.

III. DISTRIBUTED UDDI SYSTEM MODEL

A. *The architecture model*

The distributed UDDI system is strictly following the traditional UDDI standard, but the internal implementation mechanism of traditional UDDI can be improved. Therefore, a three-layer structure of distributed UDDI interoperable model is designed, including UDDI-ROOT, subsidiary UDDI and registered users, as shown in figure 4.

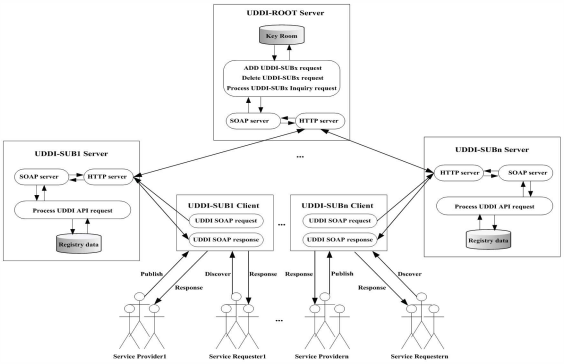

Fig.4 The design of distributed UDDI system architecture

In order to reduce the load of UDDI-ROOT Server, a UDDI-ROOT Server as root UDDI does not provide Web Services with publication and inquiry functions. UDDI-ROOT Server mainly records all relevant information of subsidiary UDDI and provides functions of registration and inquiry for the new subsidiary UDDI. Countless subsidiary UDDI, any of subsidiary UDDI-SUBn (the UDDI-SUBn represents one of the subsidiary UDDI, n represents the number) composes by a UDDI-SUBn Server and a UDDI-SUBn Client, which are mainly responsible for providing functions of service publication and service inquiry for the lower user.

A large amount of web service data and the processing of service request are able to balance on distributed UDDI model, which will relieve the performance bottleneck of service register center; when one of service registration center node is failure, the registered information can be inquiry in the rest of the registry. Such a distributed design will balance service traffic, improve the performance bottleneck of centralized UDDI system and enhance the service performance of the system. At the same time, in order to avoid single point of failure and the loss of service data, the UDDI-ROOT Server unifies management and exchange information among the various UDDI-SUBn nodes, and introduces the "key" to manage each UDDI node. The key space of distributed UDDI system is shown in figure 5

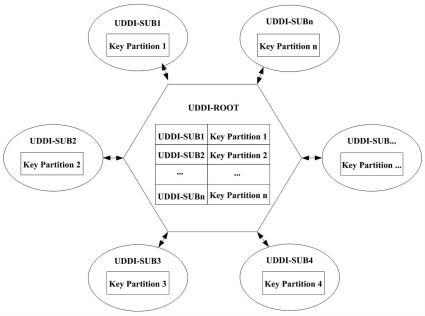

Fig.5 Key space of distributed UDDI system

UDDI-ROOT is mainly used to act as authorized agency for key space which can authorize the key partition of the UDDI-SUBn nodes, so the uniqueness and relevance of key partition of the subsidiary UDDI-SUBn nodes can be verified and maintained. When a new UDDI-SUBn node is added into or existing UDDI-SUBn node is failure, the key space of UDDI-ROOT Server will update the changed node key. UDDI-ROOT Server can easily achieve the processing of adding UDDI-SUBn node or deleting UDDI-SUBn node and inquiry the request as soon as the key of each UDDI node granted by the root registry.

B. *Information interaction*

Since the SOAP (Simple Object Access Protocol)has the advantage of cross-language and cross-platform and can solve problems of the interaction between the heterogeneous applications well, hence the information interaction of the distributed UDDI system is basically based on the SOAP protocol. Information interaction of distributed UDDI system is shown in figure 6

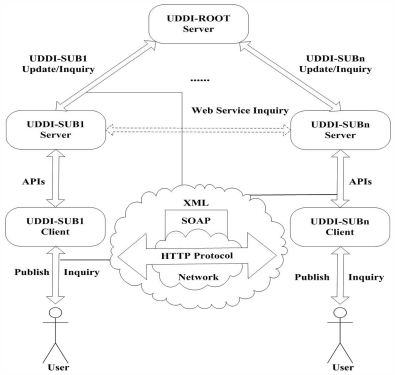

Fig.6 Information interaction of distributed UDDI system

Users log into a subsidiary UDDI-SUBn Client-side registry system, the service management module of UDDI-SUBn Client packages the encapsulated request into the SOAP message of XML format, with the opening publish or inquiry the Web Service, and then transmits through the HTTP protocol. At the same time, the UDDI-SUB1 Client calls the APIs of UDDI-SUB1 Server node or the UDDI-SUB1 Server initiates inquiry service request to the UDDI-ROOT Server. UDDI-ROOT Server also transmits via HTTP protocol SOAP request and replies packets.

IV. IMPLEMENTATION OF DISTRIBUTED UDDI SYSTEM

A. *The identity authentication*

As the prerequisite for the interaction with the registry server, an authentication token is essential for users to get into the UDDI system to publish API and inquiry API. The time sequence diagram of accessing permission is illustrated in figure7.

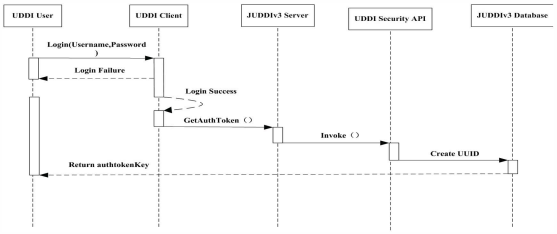

Fig.7 The timing diagram of accessing permission

UDDI users log into the UDDI client by user name and password information. After logging, UDDI Security API will be called through GetAuthToken () to verify whether it is successful. If successful, the JUDDDI database will return the AuthToken to the UDDI users.

Access token is a data of UUID type, associated with the user name and password information. AuthToken is available for a limited period, the users need to apply to the registry server for a new access token once the token expired.

B. *Service publication*

The UDDI system is the go-between between service provider and requester, so the distributed UDDI system needs to achieve the functions of service publication and service discovery to help service provider and requester establish a connection.

The operation of Service Publication mainly utilizes the released API function of UDDI to achieve the update of data structure service information which based on UDDI specification. The timing diagram of service publication operation is shown in figure 8.

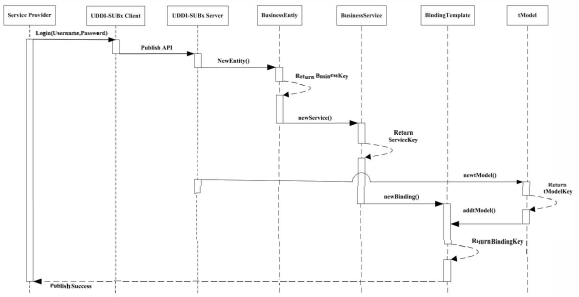

Fig.8 The timing diagram of service publication

First, Service Provider logs into the UDDI-SUBn Client by the user name and password information. Once logged successfully, user can carry out service publication. When client executes service publication, client needs to invoke the

API function of UDDI-SUBn Server and publish service information in accordance with BusinessEntity, BusinessService, TModel, BindingTemplate these four kinds of data structure sequence. When added various kinds of data type, the UDDI registry center will give a uuidKey to this data type. For updating the existed data information, use its uuidKey to invoke it, and then operate this data type.

However, due to a variety of data types are interrelated, the operation of deleting and storing cannot arbitrarily change their operational sequence. That is to say, only after publishing a BusinessEntity can it registers its BusinessService, further publish the service of BindingTemplate to add the binding between TModel and the service information, and must ensure that TModel and BusinessService have already been stored in the UDDI databases, otherwise the system will produce abnormal.

C. *Service discovery*

The operation of Service Discovery mainly utilizes the find API function of UDDI to achieve the inquiry of data structure service information which based on UDDI specification. The timing diagram of service discovery operation is shown in figure 9.

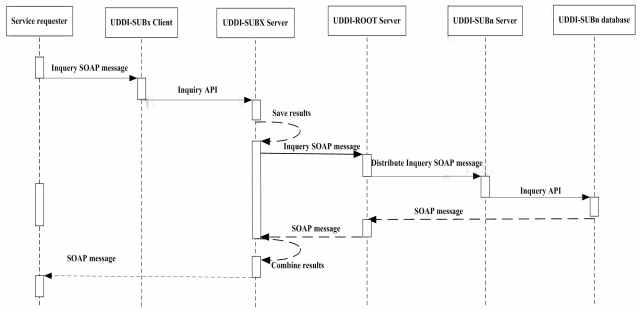

Fig.9 The timing diagram of service discovery

Each object of data type has its own name and description information, based on the information requester can inquiry these objects. Service requester inquiries service for a soap request to UDDI-SUBn Client, then UDDI-SUBn Client invokes the Inquiry API of UDDI-SUBn Server to execute the inquiry and the result set of record is returned, at the same time, inquiries service for a soap request to UDDI-ROOT Server again. According to the key information of key space, UDDI-ROOT Server distributes requests to each UDDI node, and then each UDDI node performs Inquiry API and returns the records to the root node. UDDI-ROOT Server sends the SOAP message of inquiry records to UDDI-SUBn Server, the subsidiary UDDI merges the set of result and sends it to the service requester. Such distributed UDDI system completes the strings of the service provider and service requester, and on this basis the service provider and the service requester can further complete the work of Web Service binding and calling.

D. *Service information diffusion synchronization mechanism*

Under the distributed structure, due to the presence of the cache information, the update of the service information in a single subsidiary UDDI is required to spread to the adjacent UDDI, which caches the information. As a result of the presence of synchronous update mechanism, after updating the service information, the adjacent UDDI should be informed, and then utilizes service information diffusion synchronization mechanism to implement the update of service information in the cache. Figure 10 shows the extended state chart of UDDI.

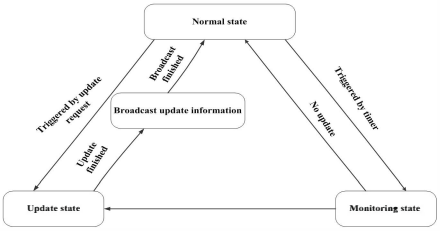

Fig.10 The extended state chart of UDDI

Normal state (normal) refers to the UDDI passively waiting for the state of the triggering event. Monitoring state (monitor) refers to the UDDI initiative to send the request for monitoring to service provider. Update state (update) refers to service provider returning the UDDI information to update the status of service registry. In the normal state, the UDDI is triggered by event timing, then into the monitoring state. If the monitored services have not been updated, the monitoring state backs to normal state; if the update information needs to be updated, through the triggered updating information into the update state. After updating, broadcast the update information to the adjacent subsidiary UDDI and then back to normal state. UDDI also receives update request of the service provider and triggers by the update request to enter the update state. After updating, broadcast the update information to the adjacent subsidiary UDDI and then back to normal state.

E. *System effectiveness analysis*

The UDDI system based on SOA needs good stability. When the number of users to access is substantial increasing, according to the release of UDDI and the inquiry of UDDI, the stability of the UDDI system based on SOA is made the following tests, further analysis the performance of the system.

1) Requests for a substantial increase in service publication:

In a centralized UDDI system, the UDDI node provides services advertised capabilities. While the requests for service publication increases by 20 thousand times per second, the node's CPU utilization is volatile (mean values of multiple test results). The centralized UDDI system CPU occupancy rate for service publication is shown in figure 11.

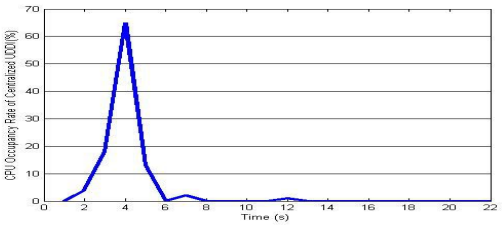

Fig.11 Centralized UDDI system CPU occupancy rate for service publication

In the distributed UDDI system, configures two UDDI-SUBn nodes to provide services advertised function, when the entire system is distributed the requests for service publication increasing by 20 thousand times per second, the node's CPU utilization is not obvious (mean values of multiple test results). The distributed UDDI system CPU occupancy rate for service publication is shown in figure 12.

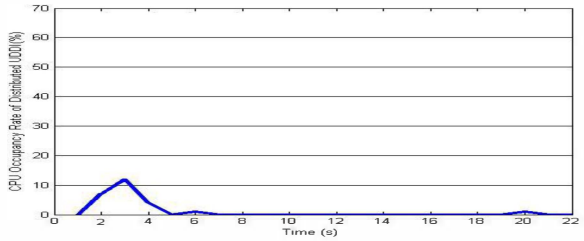

Fig.12 Distributed UDDI system CPU occupancy rate for service publication

2) Requests for a substantial increase in service inquiry:

In the centralized UDDI system, the UDDI nodes release the request for service inquiry. When the requests for service inquiry increase by two thousand times per second, the service inquiry returns the probability of the record set. Figure 13 shows the centralized UDDI inquiry performance analysis.

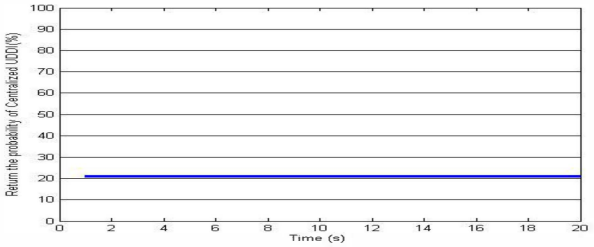

Fig.13 Centralized UDDI inquiry performance analysis

In the distributed UDDI system, the UDDI-SUB1 node releases the request for service inquiry. When the requests for service inquiry increase by 2 thousand times per second, the service inquiry returns the probability of the record set (the centralized UDDI systems uses the same query condition test) as shown in figure 14.

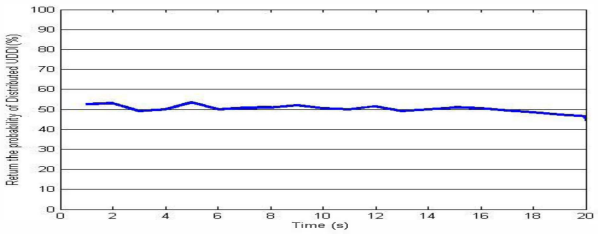

Fig.14 Distributed UDDI inquiry performance analysis

From the above shows that a substantial increase in the number of users accessing, the release performance of the distributed UDDI system based on SOA is very stable, and the inquiry performance is also very comprehensive. Synthetically the above system performance test and analysis, the distributed UDDI system based on SOA has certain service stability, effectively verifies the distributed UDDI can not only solve the problem of service publication and inquiry, but also has better technical advantages than the centralized UDDI system.

V. CONCLUSIONS

UDDI technology plays a vital role in the implementation of SOA, however, the limitation of traditional centralized UDDI leads to the bottleneck to development. Therefore, the design of the distributed UDDI is proposed, and further integration of Web Services technology protocol standards related to Java is as the implementation platform. Based on the platform of Java, the simulation system of distributed UDDI registry center is implemented, including the main functions of the publication and inquiry, and verifies the implementation of the application of distributed UDDI system.

REFERENCES

[1] Xie Chunli, Yu Ximeng, Wang Shuqin. Reliability Prediction Approach for Web Service Based on SOA [J]. Computer Science, pp. 222-226, July 2014.

[2] Jiamao Liu, Ning Gu, Yuwei Zong. Service registration and discovery in a domain-oriented UDDI Registry, Computer and Information Technology , vol 31, pp.276-283, December 2005.

[3] Wu Feng, Xu Yuelei. Research and optimization on UDDI service registry [J]. Modern Electronics Technique, vol 34, pp. 1-10, June 2011.

[4] Sean Kennedy, Robert Stewart, Paul Jacob. StoRHm: a protocol adapter for mapping SOAP based Web Services to RESTful HTTP format[J].Electronic commerce research,pp.245-269,November 2011.

[5] Zhang Yanqin. A Data Consistency Optimized Model in Distributed UDDI [J]. Fire Control & Command Control, pp. 10-19, May 2012.

[6] Long Yunjian, He Qian, Wang Yong, Wang Xiaofeng. Peer-to-peer based Web service registration system [J]. Computer Applications, pp. 1983-1987, July 2014.

[7] Yue Dai, Yongxin Feng, Yuntao Zhao, Yingchun Huang. A Method of UDDI Service Subscription Implementation[C].Proceedings of 2014 IEEE 5th International Conference on Software Engineering and Service Science, pp.661-666 , June 2014.

[8] Liang Yuanning, Chen Zhe, Xie Lijun. LIANG Yuanning, CHEN Zhe, XIE Li-jun [J]. Application Research of Computers, pp. 955-960, March 2012.

[9] Celik Duygu,Elci Atilla.A broker-based semantic agent for discovering Semantic Web services through process similarity matching and equivalence considering quality of service[J].Science China Information Sciences, pp. 1-24, January 2013.

[10] Clement L, Hately A, Riegen C V, et al, Universal description discovery &integration (UDDI) 3.0.2[EB/OL], http://uddi. org/ pubs/uddi-v3.htm, 2004.

[11] Dong Meng, Jienan CAO, Peidong Zhu. Trust management for Web Services based on extended UDDI [J]. Computer Engineering & Science, pp. 258-264, February 2014.

[12] Zhang Feng-jun, Chen Yulei. Application Analysis on Service Published and Discovery Technology in SOA [J]. Communications Technology, pp. 1990-1994, March 2014.