A Structure P2P Based Web Services Registry with Access and Control

He Qian, Zhao Baokang, Long Yunjian, Su Jinshu, Ilsun You

A Structure P2P based Web Services Registry With Access and Control

He Qian[1,2], Zhao Baokang[1], Long Yunjian[2], Su Jinshu[1], Ilsun You[3]

1. College of Computer, National University of Defense Technology, Changsha 410073
2. Key Laboratory of Cognitive Radio and Information Processing, Ministry of Education, Guilin University of Electronic Technology, Guilin 541004, China
3. Korean Bible University, South Korea

Abstract: In the cloud computing, there are massive functions and resources are encapsulated into Web services. The traditional web service registry systems normally using the central architecture can't meet the requirements of cloud computing. A web service registry system based on structured P2P system with secure access and control is implemented. Multiple Universal Description, Discovery and Integration (UDDI) nodes are organized by the P2P based schedule and communication mechanism. The registration and discovery of Web services is redesigned to the new system that provides services like one single UDDI server. The experiment results show that the capacity can be extended dynamically and support large scalable access.

Keywords. Web Service Registry, UDDI, P2P, Pastry, Access and Control

1. Introduction

The cloud computing is separated into three layers: Infrastructure as a Service (IaasS), Platform as a Service (PaaS) and Software as a Service (SaaS). Service is a main form provided by the cloud, especially in SaaS. The Service Oriented Architecture (SOA) is a new type of mode for software development, deployment and integration, which is behind object-oriented and components based development [1][2]. Flexible design and development program provided for software development is the main way of the cloud computing to provide access to the outside. SOA consists of three roles, including service provider, service registry center and service requestor. The service registry center is the basis for the realization of service composition among the entire SOA. UDDI (Universal Description, Discovery and Integration) is a descriptive specification for information related Web service, and it also includes the standardized specifications of Web services information registry center at the same time [3].

The centralized architecture and complies with the private service registry library of the UDDI specification is normally used in traditional service registration systems [3][4]. In the cloud computation, there are massive functions and resources are encapsulated into Web services. Because of the shortcomings like performance bottlenecks, single-point-of-failure and no easily scalability, which centralized architecture sys-

tems have, the traditional service registration systems are almost unable to adapt to large-scale service registration and inquiries [5-10]. Architecture for semantic sensor matchmaker was proposed by the authors to make the discovery and integration of web services more efficient. A kind of Web service discovery method called I-Wander which improves efficiency is designed for the unstructured P2P network in literature [9]. However, the service query mechanism of unstructured P2P network is completed by flooding, which has certain blindness and encroaches on a lot of network bandwidth. For the low routing efficiency, poor scalability, load imbalance and other issues which unstructured P2P network have, a service discovery method based on the structured P2P network is proposed, taking advantage of a structured P2P system to use information routing between nodes instead of flooding mechanism. With the development of P2P technology, the P2P networks can provide appropriate exchanging mechanism between the private service registration libraries, avoiding problems like solitary-island service, and it becomes a research tendency to utilize the P2P network's advantages to solve problems like performance bottlenecks of centralized architecture and single-points-of-failure [11]. The introduction of distributed architecture is proposed, using a distributed architecture to reduce the burden of registration centers, improving system efficiency. In response, a service discovery method based on the structured P2P network is proposed, taking advantage of a structured P2P system to use information routing between nodes instead of flooding mechanism. JUDDI is an very famous open source project which realizes UDDI functions [4].

This paper make a scalable architecture to extend JUDDI system, to solve some problems as performance bottlenecks, single-points-of-failure for the cloud computing. Pastry [12], which is a structured P2P protocol, is introduced to organize and coordinate multiple JUDDI. The structure P2P based Web services registration is called PUDDI in the next, and the discovery is redesigned to PUDDI that works like a single JUDDI and can provide a scalable services organization functions for big data analysis. The scalability and performance of PUDDI system is proved and analyzed through the experiments using SoapUI [13] and LoadRunner [14].

The rest of this paper is organized as follows. The framework is proposed in Section 2. The system scheduling and communication algorithm based on Pastry is given in Section 3. The registration and discovery of web service on PUDDI is presented in Section 4. The performance is evaluated in Section 5. Finally, we conclude the paper in Section 6.

2. Framework of P2P based service registration system

The Pastry protocol is introduced to organize multiple UDDI to realize a scalable Web services registration. According to the characteristics of P2P protocols, on the design of system scalability, the number of nodes can be arbitrarily expanded, the system has good scalability and strong expansibility, in other words, the system performance does not decrease with the expansion of nodes and it has a good stability; on the performance of load balancing of the network, each node on the Pastry network only keeps resources managed by itself, network services is distributed substantially

and uniformly to each node, and this is conducive to the network load balancing , increasing system throughput, supporting more concurrent users to access; on the reliability of the system, the service request is automatically transferred to the adjacent nodes by Pastry network to avoid a single-point-of-failure when there is a large number of service requests focus on a particular node or a sudden failure of a node. Based on the above advantages, Pastry agreement is introduced to take advantage of the Pastry network to improve the dynamic expandability, load balancing and other properties of the system. The network architecture of Pastry-based Web services registration system is diagrammed in Figure 1. Service requestor can send queries to any UDDI node on PUDDI, the root node is the first node to establish Pastry, based on the Pastry protocol, the routing and communication can be implemented between nodes.

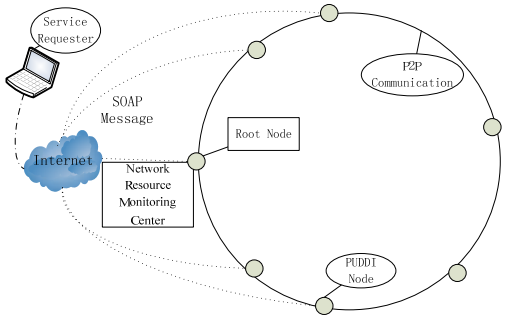

Fig. 1. The network architecture of PUDDI system

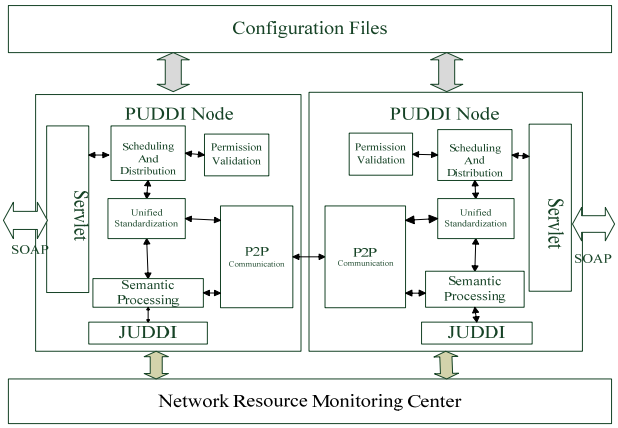

Fig. 2. The function structure of PUDDI

The functional modules of PUDDI system consists of six modules, including configuration, scheduling and distribution, peer to peer communications, access and control, JUDDI and network resource monitoring, the models are shown in Figure 2. To start Pastry, firstly, PastryIdFactory is called to hash the public key to get node ID, if

there is no public key, the IP address will be hashed instead. The P2P based scheduling module is mainly used to intercept service request messages, controlling operational processes. The P2P communication module includes sending module and reception module. JUDDI module mainly provides service registration query of nodes. The Access control module realized the Web service authentication, which is the premise of Web service access control and the security call, permission validation module promises the PUDDI system access control and the security call, on the other hand, this module gives safety certification to the users' identity and operation, if the user has not been given relevant rights, the operation will be terminated and an error message returns. If the authentication passed, related operations can be continued.

3. System scheduling and communication algorithm based on Pastry

System scheduling algorithm and system communication algorithm are the essential algorithm to implement the combination of JUDDIV3 and Pastry. Scheduling algorithm is responsible for distributing tasks, controlling processes and coordinating multiple service requests of UDDI node. System communication algorithm is the key point to implement the exchange of service information between UDDI nodes and it is also the link between the UDDI nodes.

(a) System Scheduling Algorithms

System scheduling algorithm and system communication algorithm are the essential algorithm to implement the combination of JUDDIV3 and Pastry. Scheduling algorithm is responsible for distributing tasks, controlling processes and coordinating multiple service requests of UDDI node. System communication algorithm is the key point to implement the exchange of service information between UDDI nodes and it is also the link between the UDDI nodes.When the service request messages is sent over the P2P network, the local JUDDI interface is called directly and do corresponding operations of service registry queries. If it is the reply message sent over P2P networks, the message will be returned to the client. If it is the SOAP messages initiated directly by the client, executing purview certification, if the authentication fails, an error message returned to the client. If authenticated, the service will be returned directly if the service in the query buffer, otherwise taking out the information in the message which is named tmodelName, generating a message keyword K by the SHA-1 algorithm, the K is compared with the node ID of UDDI. After that, waiting for the message reply, the reply message will be returned to the client if it is received, if the wait times out, then a times out message is returned, if the wait does not time out, then continue to wait.

The Pseudo-code of the specific process of the system scheduling algorithm is described as following, in order to a convenient introduction; the following notation is given as the algorithm 1.

Algorithm 1: Pastry based system scheduling.

```
K: The keyword of the service request message.
NodeId: The ID number of the home node.
1) if isPastrySentMessage then
2)     Invoking JUDDI API;
3) if isPastryBackMessage then
4)     if timeout then {Putting into register; }
5)    else {Return to client; }
else
{
6)  if isPassed {
7)      if storedInRegister then
8)          Return storedMessage;
else
{
9)      StandardWord ← getStandardWord(Message.tModelName)
10)      K ← SHA-1(StandardWord);
11) if mostSimilar then
12)        Invoking JUDDI API;
else
{
13)        Create PastrySentMessage;
14)        Invoking Pastry API;
15)          Wait;
16)        if receivedPastryBackMessage then
17)          Return to client;
18)        else if timeout then
19)           Return timeoutMessage;
20)        else Goto (12);
}}}
else //the authentication does not pass;
20)    Return error;
}
```

(b) System communication algorithm

The specific process of system communication algorithm is: firstly, the service messages which need to be sent stored into the message buffer, the keyword K of the message is compared and matched to the node ID of UDDI by the Selection manager, (the service messages) will be forwarded to the UDDI node which exists in the UDDI node set which close to the home node, and the condition is, compared to home node, this UDDI node is more closer to the keyword K. If the sending is successful and

ACK is received, the next service message will be sent, otherwise there is a retransmit. If the node cannot be found, then the length m of the common prefix which both the ID number of the home node and the keyword of the message have is calculated. If the node ID of UDDI exists in m rows and n columns (i is the value of number m in K) in the routing table, forwarding the service message to the UDDI node, if the node does not exist, the message is forwarded to the node which the length of common prefix is m, and compared to the home node, this node is closer to K.

The Pseudo-code of the specific process of the system communication algorithm is described as following, in order to a convenient introduction, the following notation is given as algorithm 2.

Algorithm 2: Pastry based communication.

K: Keyword of the Pastry message

NodeId: The ID number of home node.

R_l^i: The UDDI node exists in l ($0 \le l < 128/b$) rows and **오류!**

참조 원본을 찾을 수 없습니다. ($0 \le i < 2^b$) columns in the routing table。

L_i: The i-th ($-|L|/2 \le i \le |L|/2$) UDDI node in the UDDI set, L ,which is close to the home node.

K_l: The numerical value in the l-position of the keyword K of the service request message。

1) Put Message into buffer;

2) SelectorManager.Match(*K*);

3) if $L_{-|L|/2} \le K \le L_{|L|/2}$ then

{

Forward $UDDI_L$, ($UDDI_L \in L \cap \min(|L_i - K|)$);

5) if *ACK*==true then Goto (1);

6) else Goto (2);

}

else //forwarding by using of routing table;

{

7) l =prefix(*K,nodeId*);

8) if $R_l^{K_l} \neq NULL$ then

{

9) Forward $UDDI_R$, ($UDDI_R == R_l^{K_l}$);

10) if *ACK*==true then Goto (1);

11) else Goto (2);

}else //this UDDI node does not exist {

12) Forward $UDDI_R$, ($UDDI_R \in R \cap \min(|R_l^i - K| < |nodeId - K|)$);

13) if *ACK*==true then Goto (1);

14) else Goto (2);

}
}

For example, a message whose key is D617FA is sent to the node whose ID is 74BA2F, according to the system communication algorithms, the forwarding process of the message is shown in Figure 3. After each forwarding message between the nodes, the message is more close to the target node, and finally it will be forwarded to the node which ID is the closest to the K value, and the ID is D615AB.

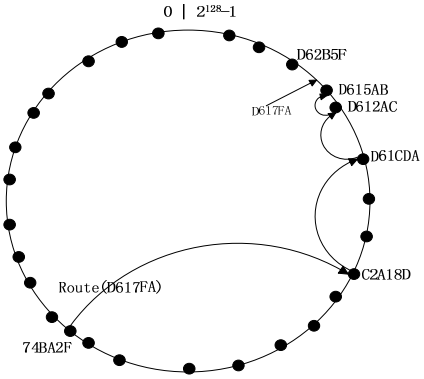

Fig. 3. Routing process of the system message

4. The registration and discovery of web service based on PUDDI

The service provider is responsible for the concrete implementation of the service, and it is also responsible for publishing the service to the registration center. The service requester can find service description in the registration center and obtain binding information, utilizing the binding information to bind to the service provider, and then call the service provided.

The concrete processes that service providers register service to the PUDDI is shown in Fig.4, and described as following.

Step 1: the service providers read the URL(Uniform Resource Locator) of the services' WSDL(Web Services Description Language) file and parse the WSDL file, generating the necessary information when there is service registration (For example, binding information).

Step 2: The related messages of SOAP are generated and the messages will be sent to a known node on the P2P networks through the http requests

Step 3: After the messages of SOAP are received by the node, these messages are transferred from the schedule and distribution module to the access and control module, and operate. According to the results returned by the schedule and distribution module, the purview certification module gives it's judgment, if the purview certifica-

tion is not passed, an error message is returned, if the purview certification is passed, the next process continues.

Step 4: Analyzing the service-related information, a hash value is obtained through SHA-1 algorithm; According to the hash value, the node judges whether the messages are managed by itself, if so, then jump to the step f, if not, the message of SOAP will be packaged through Pastry protocol and transferred to the P2P communication module, and then the messages will be forwarded to the node which manage the hash value.

Step 5: According to the routing mechanisms of Pastry, the messages will be finally forwarded to the node who manages the services, by P2P communication module. Then the messages will be forwarded to and handled by the schedule and distribution module

Step 6: Calling the registered API of JUDDI, the analyzed information of the WSDL file of service and some related information will be registered to the JUDDI of this node.

Step 7: Returning the registered information of the result to the client.

The process of service registration is shown in Fig.5. The procedure that the service requester query to the PUDDI of the founding service is similar to the procedure of registration and the difference is that the initiated message of request is not the same and no more description here.

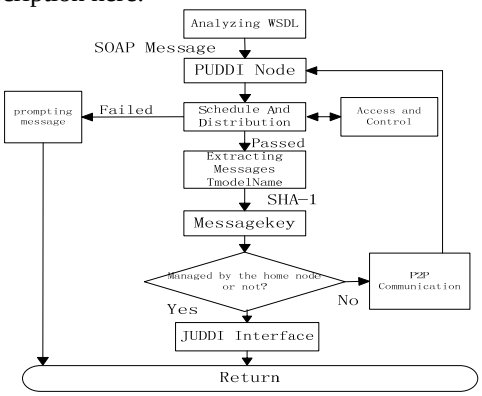

Fig. 4. Process of service registration

5. Experiments and analysis

The experiment is tested in the network environment of 100M Ethernet, including 3 physical hosts and 3 virtual machines. The physical hosts are Lenovo foolproof server of R510 (Intel 4-core, 2.80GHz processor, 1G RAM), the 3 virtual machines (single CPU, 1G RAM) are deployed on the Sugon server of W5801-G10 (two Intel 12-core, 2.30GHz processor, 64G memory).The operating systems are centos 6.2 and the software environment includes the Java platform of jdk1.6, database of mysq15.5 and web container of Apache Tomcat6.

The SoapUI is used to test the system function of PUDDI. For the query of Web service, the service requestor firstly queries the WSDL address of the service, and then generates a client based on the WSDL, after that, you can call the service or even combine new service. All the function works like a single JUDDI. The performance of a software system is measured by some common indicators: response time, throughput, the number of concurrent users. The response time is the time which spent on the service which the software system provide for the user, and the response time includes the response time of server, the response time of internet and response time of client. The response time can intuitively reflect the processing capacity of the system, so, according to the response time, we can measure the performance of the JUDDUV3 system and the PUDDI system. Next, The LoadRunner is used to test the service registration performance and the discovery performance of the JUDDI system and the PUDDI system which deploys 6 nodes.

(a) Web services registration

In the experiment, for the original system of JUDDIV3 and the PUDDI of 6 nodes, loadRunner is used to simulate the service registration of multiple users. 5 users is added in every 10 seconds until the concurrent users reaches to 100, and the test has to work for a continuous time. The average response time of the service registration of JUDDIV3 and PUDDI is shown in figure 5 and table 1. According to the chart, the response time of service registration increases as the number of the concurrent users increases, however, the increased magnitude of the response time of JUDDIV3 system is far greater compared to PUDDI system, and the response time of registration of the JUDDIV3 system is twice time more than PUDDI system. The standard deviation of PUDDI system is much smaller than JUDDIV3 system, this illustrates that the impact of the PUDDI system, which suffered by the increasing users, is small, and the system is more stable.

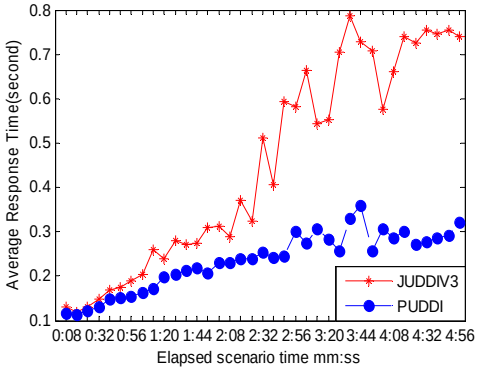

Fig. 5. The contrastive chart of average response time

Table 1. the contrastive table of the response time of registration（second）

Test objects	Average	Minimum	Maximal	Median	standard

	value	value	value	value	deviation
JUDDIV3	0.451	0.12	0.787	0.405	0.23
PUDDI	0.234	0.113	0.36	0.243	0.064

(b) Web Services discovery

In this experiment, the service of JUDDIV3 system and PUDDI system is the same, under the circumstance that 2000 different services have been registered, 100 concurrent users are simulated and the query services of each user are random. 5 concurrent users are started in every 10 seconds until the number of the users reaches to 100, and the test has to work for a continuous time. The average response time of the service registration of JUDDIV3 and PUDDI is shown in figure 6 and table 2. According to the result of the experiment, the response time of service discovery increases as the number of the concurrent users increase, however, the time of service discovery of the JUDDIV3 system is twice more than PUDDI system. The standard deviation of PUDDI system is much smaller than JUDDIV3 system, this illustrates that the impact of the PUDDI system, which suffered by the increasing users, is small, and the system is more stable.

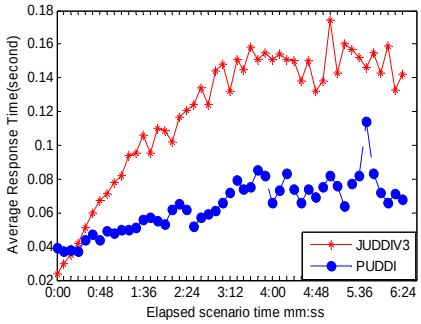

Fig. 6. The contrastive figure of response time of queries

Table 2. The contrastive table of response time of queries (second)

Test objects	Average value	Minimum value	Maximum value	Median value	Standard deviation
JUDDIV3	0.12	0.024	0.174	0.134	0.039
PUDDI	0.064	0.037	0.114	0.066	0.015

The clicks of PUDDI system and JUDDIV3 system while querying the service is shown in Figure 7. In figure 7, X-axis represents the running time; Y-axis represents the clicks per second. The clicks reflect the number of SAOP requests which initiated by the clients, and this reflects the processing ability of the system from the upper side, the more strong the processing ability of the system, the more requests sent from the client will be received. According to the Figure 7, the clicks of PUDDI system is far larger than JUDDIV3 system, and the processing ability of PUDDI system is stronger. The contrastive result of the experimental data is shown in Table 3.

Table 3. The contrastive table of clicks of the systems

Test objects	Aver-age value	Mini-mum value	Maxi-mum value	Median value	Standard deviation
JUDDIV3	192.251	61.25	246.125	204.125	39.182
PUDDI	286.551	14.375	394.75	305.5	88.68

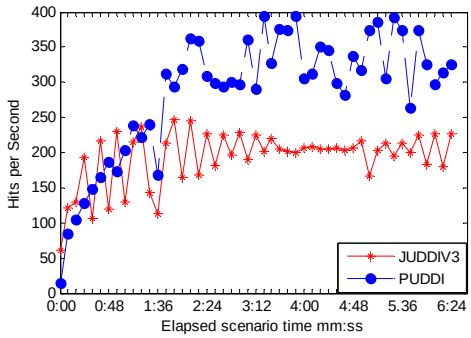

Fig. 7. The contrastive figure of clicks

6. Conclusion

An extendible distributed service registration system is needed for the management of the massive web services. A service registration system based on the structured P2P is designed in this paper. The implemented Pastry based Web service registration system can support magnanimous web services registration if there are enough computers provided and it can adapt to the large-scale registration and access dynamically. PUDDI works like a Web service cloud that provides a scalable services organization capacity for the cloud computation and the big data analysis.

Acknowledgment

This work was partly supported by the National Natural Science Foundation of China (61201250,61163057, 61163058), the Important National Science & Technology Specific Projects (2012ZX03006001) and Guangxi Natural Science Foundation of China (2012GXNSFBA053174).

Reference

1. Al2Masri E , Mahmoud Q H. Investigating Web services on the World Wide Web [C]. in Proc of the 17th Int Conf on World Wide Web. New York : ACM , 2008 : 795-80.

2. M.P.Papazoglou,P.Traverso,S.Dustdar,F.Leymann."Service-Oriented Computing: A research roadmap," International Journal of Cooperative Information Systems,vol.17,pp.223-255, Jun 2008.
3. Liu J X, Liu J, Chao L. Design and implementation of an extended UDDI registration center for web service graph[C]// ICWS 2007: Proceedings of IEEE International Conference on Web Services. .Salt Lake City: IEEE Press, 2007: 1174-1175.
4. Apache Software Foundation .JUDDI.[EB/OL].[2014-01-17]. http://juddi.apache.org/.
5. TAMILARASI K, RAMARKRISHNAN M. Design of an intelligent search engine-based UDDI for web service discovery[C]//ICRTIT 2012: Proceedings of International Conference on Recent Trends In Information Technology. Chennai : IEEE Press, 2012: 520-525.
6. Yao Y, Cao J X, LIU B, et al. Scalable mechanism for semantic web service registry and discovery[J]. Journal of Southeast University (Natural Science Edition), 2010, 40(002): 264-269.
7. Guo M Q, Huang Y, Luo X G, et al. Design and implementation of a distributed web service directory in SOA-oriented urban spatial information sharing platform[C]// PACIIA 2009: Proceedings of Asia-Pacific Conference on Computational Intelligence and Industrial Applications.. Wuhan: IEEE Press, 2009, 1: 127-130.
8. J. C. Goodwin, D. J. Russomanno, J. Qualls, Survey of Semantic Extensions to UDDI: Implications for Sensor Services, SWWS, 2007 ; 16-22.
9. Zhang C Y, Cao Y, Liu D, et al. I-Wander: A Web Service Discovery Method for Unstructured P2P Network[J]. Transactions of Beijing Institute of Technology, 2008, 26(6): 521-525. Wander:
10. Sioutas S, Sakkopoulos E, Makris C, et al. Dynamic Web Service discovery architecture based on a novel peer based overlay network[J]. Journal of Systems and Software, 2009, 82(5): 809-824.
11. R.Steinmetz,K.Wehrle. Peer-Peer-NetworkingComputing. Informatik Spektrum, 2004,27(1): 51-54.
12. A.Rowstron, P.Druschel,. Pastry: scalable, decentralized object location and routing for large-scale peer-to-peer systems. In: 18th IFIP/ACM Conference on Distributed Systems Platforms, Heidelberg (D), 2001: 86-94.
13. Luo Z M, Zhu Y, Cheng M. Web service testing tool soapui and its analysis[J]. Computer Applications and Software, 2010, 27(005): 155-157.
14. Mercury Interactive,LoadRunner[EB/OL]..http : //www.mercury.com/us/products /loadrunner/, 2007.