ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Decentralized Service Discovery Approach on Peer-to-Peer Networks

Gumma.Parvathi Devi
M.Tech Student,
Dept of CSE,
VITS College.

N.Ramanjaneya Reddy
Asst.prof,
Dept of CSE,
VITS College.

ABSTRACT:

Service-Oriented Computing (SOC) is a well known application in distributed computing. The issues related to SOC have to learn the recent mechanism to overcome the security threats. But SOC using centralized registries can easily suffer from bottleneck and vulnerability problem which leads to fail.

To overcome this process , this paper proposes a peer-to-peer-based decentralized service discovery approach named Chord4S which utilizes the data distribution and lookup capabilities of the popular Chord to distribute and discover services in a decentralized manner. The experimental analysis shows the Chord4S achieves higher data availability and provides efficient query with reasonable overhead.

Key words:

SOC, Peer net works, Chord4S.

Introduction:

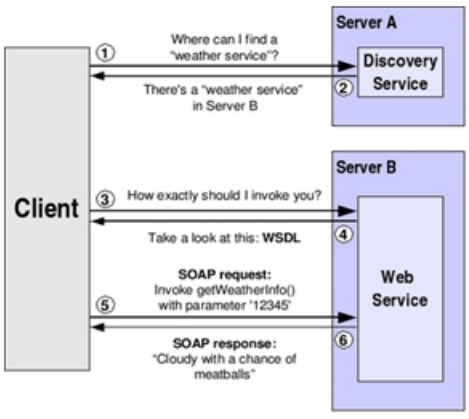
A web service is a method of communication between two electronic devices over the World Wide Web. A web service is a software function provided at a network address over the web or the cloud, it is a service that is "always on" as in the concept of utility computing.

The W3C defines a "Web service" as software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

How Web Services Works?

The basic Web services platform is XML + HTTP.
XML provides a language which can be used between different platforms and programming languages and still express complex messages and functions.
The HTTP protocol is the most used Internet protocol.
Web services platform elements:

- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)



Structure of Web Services

Web Services are all about XML:

1.A Web Service is described by a document in XML format, in the XML language known as WSDL (Web Services Description Language). This describes the service in terms of the operations, messages, and bindings that it contains, and may provide a URL at which the service may be called.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

2.The structure of the messages is described using XML Schema (XSD) which is either contained in, or referred to by, the WSDL.

3.The messages sent to, and received from, the web service are all in the form of XML that complies to the schema, and which follows the protocols described by the WSDL, using an XML protocol known as SOAP (for Simple Object Access Protocol)The usual way to access a web service in .NET is as follows. Other platforms have similar mechanisms.

1.The developers of the web service make the WSDL accessible. They can either place it on a web site, or the web service can generate the WSDL automatically.

2.The developer of a client application will add a Service Reference if using WCF, or a Web Reference if using the older ASMX technology. This amounts to telling Visual Studio the location of the WSDL for the web service.

3.Visual Studio requests the WSDL. It parses the file to gain an understanding of the web service. From this information, it will create some classes. These are known as proxy classes (not to be confused with an Internet proxy server). These classes stand in for the web service, allowing you the illusion that you are simply calling normal methods on normal classes. But these classes are not normal.

4.The client code will create an instance of the proxy class for the service. It will call an instance method of the class, perhaps passing some parameters. The proxy class will turn these parameters into properly-formatted SOAP XML and will send them to the service.

5.On the server side, the XML will likely be turned into parameters again, and a server-side method will be called with those parameters.

6.The server will do what it was requested to do, and may then return a result. This result will be turned into SOAP XML, which will be sent back to the client.

7.The proxy class on the client will receive this XML, turn it back into return values and will return that to the caller

The result is that the caller of a web service can treat the service operations just like normal methods on normal objects. In most cases, even the developer of the web service can treat the service like it is a normal object with normal methods. But everything in the middle is happening in SOAP XML, described by XML Schema, referenced by a WSDL in XML.

Characteristics of Web Services:

Machine-to-machine interactions
Loose coupling
Interoperability
Platform-independence
Operating system-independence
Language-independence
Leveraging the architecture of the World Wide Web
Benefits of Web Services:
Web Services offer many benefits over other types of distributed computing architectures.

Interoperability - This is the most important benefit of Web Services. Web Services typically work outside of private networks, offering developers a non-proprietary route to their solutions. Services developed are likely, therefore, to have a longer life-span, offering better return on investment of the developed service. Web Services also let developers use their preferred programming languages. In addition, thanks to the use of standards-based communications methods, Web Services are virtually platform-independent.

Usability - Web Services allow the business logic of many different systems to be exposed over the Web. This gives your applications the freedom to chose the Web Services that they need. Instead of re-inventing the wheel for each client, you need only include additional application-specific business logic on the client-side. This allows you to develop services and/or client-side code using the languages and tools that you want.

Reusability - Web Services provide not a component-based model of application development, but the closest thing possible to zero-coding deployment of such services. This makes it easy to reuse Web Service components as appropriate in other services. It also makes it easy to deploy legacy code as a Web Service.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Deployability - Web Services are deployed over standard Internet technologies. This makes it possible to deploy Web Services even over the fire wall to servers running on the Internet on the other side of the globe. Also thanks to the use of proven community standards, underlying security (such as SSL) is already built-in.

PROJECT SCOPE:

Service Oriented Computing (SOC) is emerging as a new paradigm for developing distributed applications. Service discovery, among the most fundamental elements of SOC, is critical to the success of SOC as a whole. Traditional service discovery approaches of the web services technology are based on Universal Description, Discovery, and Integration (UDDI).

However, centralized service registries used by UDDI may easily suffer from problems such as performance bottleneck and vulnerability to failures as the number of service consumers and requests increase in an open SOC environment.

This inherent disadvantage prevents web services from being applied in large scalable service networks. As SOC environment is largely distributed, a decentralized approach appears to be the most natural way to address the above issues and achieve scalable, reliable and robust service discovery.

The Peer-to-Peer (P2P) technology provides a universal approach to improving reliability, scalability, and robustness of distributed systems by removing centralized infrastructures.In areas such as file sharing, Voice over Internet Protocol (VoIP) , and video streaming, P2P has achieved great success. Very recently, innovative research has also been carried out in the SOC field to leverage P2P computing and web services for improved service discovery.

Naturally, a P2P-based decentralized service discovery approach consists of a set of distributed nodes that form a structured P2P overlay network. Although structured P2P can potentially improve the scalability of service discovery, directly applying DHT based P2P approaches to decentralized service discovery may be weak in guaranteeing the availability of published service descriptions.

EXISTING SYSTEM:

Traditional service discovery approaches of the web services technology are based on Universal Description, Discovery, and Integration (UDDI). However, centralized service registries used by UDDI may easily suffer from problems in an open SOC environment. To overcome the problems The Peer-to-Peer (P2P) technology provides a universal approach to improving reliability, scalability, and robustness of distributed systems by removing centralized infrastructures. Based on Distributed Hashing Table (DHT), structured P2P systems can achieve even data distribution and efficient query routing by controlling the topology and imposing constraints on the data distribution. In this technology also problems occur.

DISADVANTAGES OF EXISTING SYSTEM:

Although structured P2P can potentially improve the scalability of service discovery, directly applying DHT based P2P approaches to decentralized service discovery may be weak in guaranteeing the availability of published service descriptions. This is because DHT-based systems often distribute descriptions of functionally equivalent services to the same successor node, as they have the same or similar hashing values. If such a node fails, a service consumer will not be able to discover any of these services. This disadvantage may result in serious problems in open and dynamic SOC environments where unexpected failure of nodes cannot be avoided.

PROPOSED SYSTEM:

This paper proposes Chord4S, a Chord-based decentralized service discovery approach that supports service description distribution and discovery in a P2P manner. Chord is selected because it is well recognized for its flexibility and scalability and is considered suitable in large scale SOC environments.

ADVANTAGES OF PROPOSED SYSTEM:

Chord4S takes advantages of the basic principles of Chord for nodes organization, data distribution and query routing.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

• The main aim of designing Chord4S is to largely improve the availability of service descriptions in volatile environments by distributing descriptions of functionally equivalent services to different successor nodes.

• In case one node fails, a service consumer is still able to find functionally equivalent services that are stored at other successor nodes. Another two features of Chord4S are to support service discovery with wildcard(s) and QoS awareness.

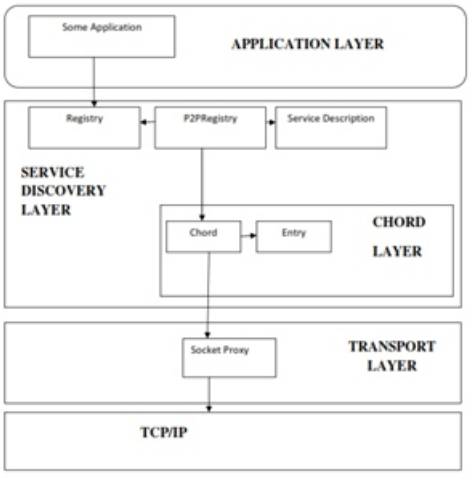
• Chord4S extends Chord's original routing protocol to support discovery of multiple functionally equivalent services at nodes with one query, which is necessary for different successor negotiation of a Service Level Agreement and selection of optimal service providers

ALGORITHMS USED:

» Algorithm1: Pseudo code for finding successor operation.

» Algorithm2: Function of match making.

SYSTEM ARCHITECTURE:



MODULES:

1. Structured Peer-to-Peer Network

2. Service Description

3. Service Publication

4. Service Query

MODULE DESCRIPTION:

1. Structured Peer-to-Peer Network:

A peer-to-peer (P2P) network is a type of decentralized and distributed network architecture in which individual nodes in the network (called "peers") act as both suppliers and consumers of resources, in contrast to the centralized client-server model where client nodes request access to resources provided by central servers. In a peer-to-peer network, tasks are shared amongst multiple interconnected peers who each make a portion of their resources directly available to other network participants, without the need for centralized coordination by servers.

2. Service Description:

The service description supported by Chord4S consists of three main parts: service identifier, QoS specification, and syntax specification. The service identifiers are the identifications of the services as the basis for routing query messages. The QoS specification specifies the quality of the service that the service provider can offer. a service identifier in Chord4S is divided into two parts, function bits and provider bits. When hashing a service description to generate the service identifier. The function bits are used for the functional service matchmaking in service discovery. The main function of the provider bits is to distinguish and distribute functionally equivalent services.

3. Service Publication:

Chord4S improves data availability by distributing descriptions of functionally equivalent services to different nodes. In this way, a failed node would just have limited impact on data availability. A service consumer has the opportunity to locate the functionally equivalent services from those available nodes. To guarantee the data availability of Chord4S-based system and application design can be consider to the Replication (i.e., storage of multiple copies of a service description at different nodes) and Redundancy (i.e., storage of redundant information along with the service description)

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

4. Service Query:

A service-specific query contains complete details of a service description and is used to look up a specific service. To initiate a service-specific query, the service consumer needs to fill in all the layers that compose the query with explicit service information. The objective of using service-specific queries is usually to look up a group of functionally equivalent services provided by different service providers. When solving a query with wildcard(s), it is actually looking up a virtual segment composed by nodes succeeding service descriptions that fall into the target service category.

CONCLUSION:

Service discovery is a critical component in service-oriented computing. Over recent years, peer-to-peer-based service discovery has attracted researchers' attention after the deficiencies of centralized service discovery are identified. This paper has proposed Chord4S, a peer-to-peer-based approach for decentralized service discovery. To improve data availability, Chord4S distributes the descriptions of functionally equivalent services.

Chord4S supports QoSaware service discovery and service discovery with wildcard(s). An efficient routing algorithm is developed to facilitate queries of multiple functionally equivalent services. Chord4S is scalable, reliable, and robust due to the enhanced peer-to-peer architecture. Experimental results demonstrate that Chord4S can achieve high data availability and efficient query of multiple functionally equivalent services with reasonable overhead. In the future, integration of semantic information of services into Chord4S using popular tools, such as Petri Net and WSMO, will be investigated in order to increase the flexibility and accuracy of the service discovery.

REFERENCES:

[1] "North American Industrial Classification Scheme (NAICS) codes," http://www.naics.com/, 2012.

[2] "Universal Standard Products and Services Classification (UNSPSC)," http://www.unspsc.org/, 2012.

[3] R. Ahmed, N. Limam, J. Xiao, Y. Iraqi, and R. Boutaba, "Resource and Service Discovery in Large-Scale Multi-Domain Networks," IEEE Comm. Surveys and Tutorials, vol. 9, no. 4, pp. 2-30, Oct.-Dec.2007.

[4] E. Al-Masri and Q.H. Mahmoud, "Crawling Multiple UDDI Business Registries," Proc. 16th Int'l Conf. World Wide Web (WWW '07), pp. 1255-1256, 2007.

[5] D. Ardagna, M. Comuzzi, E. Mussi, B. Pernici, and P. Plebani, "PAWS: A Framework for Executing Adaptive Web-Service Processes," IEEE Software, vol. 24, no. 6, pp. 39-46, Nov./Dec. 2007.

[6] S. Baset and H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol," Proc. IEEE INFOCOM, pp. 1-11, 2006.

[7] J. Beatty, G. Kakivaya, D. Kemp, T. Kuehnel, B. Lovering, B. Roe, C. St.John, J. Schlimmer, G. Simonet, D. Walter, J. Weast, Y. Yarmosh, and P. Yendluri, "Web Services Dynamic Discovery (WS-Discovery)," http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf, 2005.

[8] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, and L.A. Stein, "OWL Web Ontology Language Reference," http://www.w3.org/TR/owlref, 2004.

[9] Z. Cheng, M.P. Singh, and M.A. Vouk, "Verifying Constraints on Web Service Compositions," Real World Semantic Web Applications, June 2002.

[10] L. Clement, A. Hately, C. von Riegen, and T. Rogers, "UDDI Version 3.0.2," OASIS, http://www.uddi.org/pubs/uddi_v3.htm, 2004.

[11] F. Emekc¸i, O.D. Sahin, D. Agrawal, and A.E. Abbadi, "A Peer-to-Peer Framework for Web Service Discovery with Ranking," Proc. IEEE Int'l Conf. Web Services (ICWS '04), pp. 192-199, 2004.

[12] H. Foster, S. Uchitel, J. Magee, and J. Kramer, "Model-Based Verification of Web Service Compositions," Proc. IEEE 18th Int'l Conf. Automated Software Eng. (ASE '03), pp. 152-163, 2003.