OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

A Chord based Service Discovery Approach for Peer- to- Peer Networks

Yaminipurnatilak Jakka[1], A. Sudarsan Reddy[2]

[1] M. Tech (CSE), Vasireddy Venkatadri Institute of Technology, Guntur, A.P., India.

[2]Assoc. Professor, Dept. of Information Technology, Vasireddy Venkatadri Institute of Technology, Guntur, A.P., India

Abstract: *The limitations of client- server systems become evident in the large scale distributed environments. Peer to peer (P2P networks can be used for improving communication process, optimizing resources discovery/localization, facilitating distributed information exchange. P2P applications need to discover and locate efficiently the node that provides the requested and targeted service. This paper proposes an approach called Chord4S, a Chord-based decentralized service discovery approach that supports service description distribution and discovery in a P2P manner. The main aim of designing Chord4S approach is to largely improve the availability of service descriptions in volatile environments by distributing descriptions of functionally equivalent services to different successor nodes. Two main features of Chord4S approach are to support service discovery with wildcard(s) and QoS awareness. Furthermore, Chord4S approach extends Chord's original routing protocol to support the discovery of multiple functionally equivalent services at different successor nodes with one query, which is necessary for negotiation of a Service Level Agreement and selection of optimal service providers.*

Keywords: *Chord, Decentralized approach, P2P network, Service discovery.*

I. Introduction

Peer-to-Peer (P2P) systems (figure 1) are distributed systems without (or with a minimal) the need of centralized control or hierarchical organization, where each node is equivalent in term of functionality. P2P refers to a class of systems and applications that employ the distributed resources to perform a critical function such as resources localization in a decentralized manner. The main challenge in peer to peer computing is to design and implement a robust distributed system composed of distributed and the heterogeneous peer nodes, located in unrelated administrative domains. In a typical P2P system, the participants can be "domestic" or "enterprise" terminal nodes connected to the Internet. Peer-to-Peer computing is a very controversial topic. Many experts believe that there is not much new in peer to peer networks. There are several definitions of P2P systems that are being used by the peer to peer community. As defined in [1], "P2P allows file sharing or computer resources and services by direct exchange between the systems", or "allows the use of devices on the Internet periphery in a non client capacity". Also, t could be defined through 3 key requirements: a) they have an operational computer of server quality, b) they have a DNS independent addressing system" and c) they are able to scope with variable connectivity. Also, as defined in [2]: peer to peer is a class of applications that takes advantage of resources-storage, cycle, content, human presence-availability at the edges of Internet. Because accessing to these decentralized resources means operating in an environment with unstable connectivity and unpredictable IP addresses. P2P nodes must operate outside the DNS system and have significant or total autonomy from central servers [1].

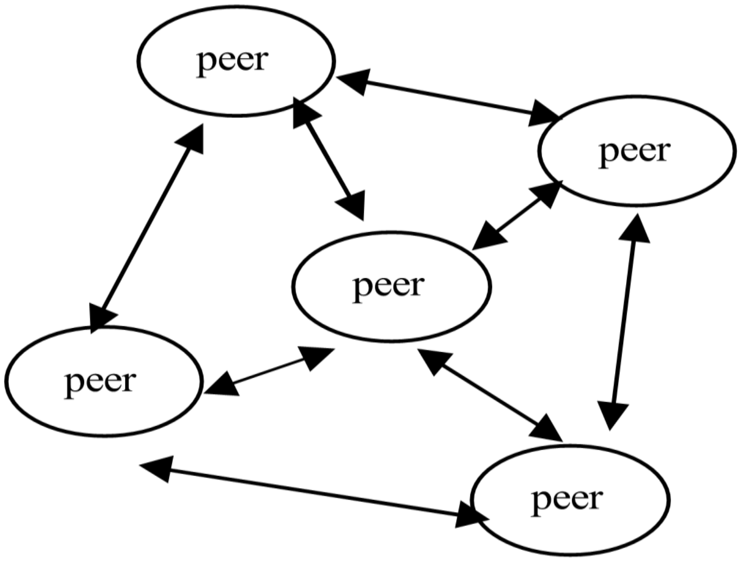


Fig. 1: Decentralized peer- to- peer Network

Naturally, a P2P-based decentralized service discovery approach consists of a set of distributed nodes that form a structured peer to peer overlay network. Upon registration, the description of a service is distributed to a relevant node to be stored in the repository. A service query can be submitted to any node, and this node, if does not store the required service description, it is able to route the query to an appropriate node for resolution. Descriptions of matched services are retrieved and returned to the service consumer as the result of the service query. This paper proposes an approach called Chord4S, a Chord-based decentralized service discovery approach that supports service description distribution and discovery in a peer to peer manner. Chord is selected because it is well recognized for its flexibility and scalability and is considered suitable in large scale service computing environments. Chord4S takes advantages of the basic principles of Chord approach for nodes organization, data distribution and query routing. The main aim of designing Chord4S approach 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 a node fails, a service consumer is still able to find functionally equivalent services that are stored at other successor nodes.

II. Related Work

The centralized client/server model has been adopted for service discovery in service oriented computing. These traditional service discovery approaches of the web services technology are based on the Universal Description, Discovery, and Integration (UDDI) [3]. The UDDI Version 3.0.2 Specification [3] describes the Web services, data structures and the behaviours of all instances of a UDDI registry. Extension for a query federation of UDDI registries within Web Service environment is depicted in [4]. The search space is enlarged and the opportunity for the service consumers to discover more services that satisfy their requirements is increased, as this allows queries for businesses or services to be forwarded transparently to those extended UDDI nodes within the federation. In [5], the authors describe an interoperable model of distributed UDDI. This model divides UDDI servers into three types: normal server, super domain server and root server. Philosophy of the Domain Name System (DNS) is adopted here. Super domain servers, which are managed by the root server, are further used to maintain normal servers. This model is exposed to the same threats that domain name system faces, e.g., Distributed Denial of Service (DDoS) attack, as it is based on concept of DNS. In [6], the authors proposed a Web Service Crawler Engine to address the performance issue caused because of huge number of UDDI registries. Required web services can be efficiently discovered from a repository by the service consumers, as the crawler in the engine crawls accessible UDDI registries and gathers information in a centralized repository.

Decentralized service discovery is considered as a promising approach to addressing the problems caused by the centralized infrastructures. In particular, some preliminary research has been conducted to utilize peer to peer computing for service discovery [7]. In [8], the authors presents an enhanced Skip Graph, Service Index, using WSDL-S as the semantic description language. Skip Graph is built by extracting semantic attributes of the web services as indexing keys. It consists of a set of increasingly sparse doubly-linked lists ordered by levels starting at level zero. In [9], the authors propose distributed web service discovery architecture. It is based on the distributed shared space concept and intelligent search among a subset of spaces. Publishing of the Web service descriptions as well as to submit requests to discover the Web service of user's interests is allowed. Integration of several applications running on different resource specific devices is also supported. In [10], the authors propose a Chord based structured peer-to-peer framework for Web service discovery in which Web services are located based on both service functionality and process behavior. Process behavior of the Web services is represented with finite automata and these finite automata are used for publishing and querying the Web services within the system. In [11], the authors propose the approach based on the concept that service providers themselves should take the responsibility to maintain their own service descriptions in the decentralized environment. To group peer nodes by service categories to form islands on the Chord ring, the decentralized service directory infrastructure is built with hashing descriptive strings into the identifiers. In [12], the authors present PSWD, a distributed web service discovery architecture based on an extended Chord algorithm called XChord [7]. In PSWD, XML is used to describe the web service descriptions and to express the service request.

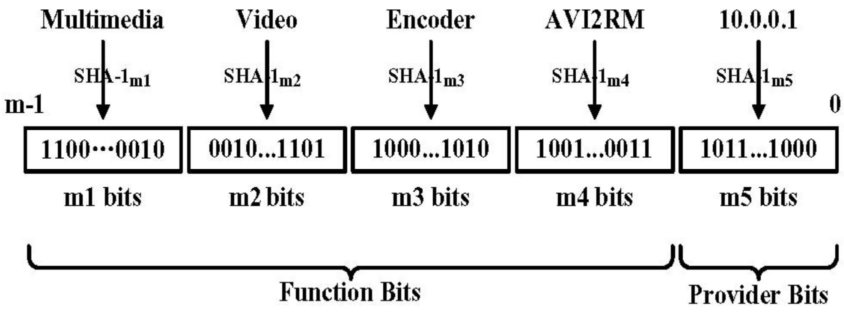
III. PROPOSED WORK

1. Chord4S Service Description

The service description supported by Chord4S approach consists of three main parts:

- Service identifier
- QoS specification
- Syntax specification.

The service identifiers are the identifications of the services as the basis for routing the query messages. A service identifier for Chord4S is divided into 2 parts, function bits and provider bits. The function bits are used to refer to the functionality of the service while the provider bits are utilized to describe provider specific information. When hashing a service description to generate the service identifier, Chord4S approach allocates certain bits of a service identifier for the function descriptions and the rest for provider bits. A sample of service identifier consisting of 5 layers is presented in figure 2.



$$\text{Service identifier}_m = hash_{m1}(\text{“Multimedia”}) + hash_{m2}(\text{“Video”}) + hash_{m3}(\text{“Encoder”}) + hash_{m4}(\text{“AVI2RM”}) + hash_{m5}(\text{“10.0.0.1”})$$

$$(m1 + m2 + m3 + m4 + m5 = m)$$

Note: Here "+" executes connection of bits.

Fig. 2: Service identifier generated from hierarchical service description

The QoS specification in Chord4S specifies the quality of the service that the service provider can offer. Chord4S approach allows service providers to publish their services with quality specifications attached as advertisements. However, the quality specifications are not involved in the generation of the service identifier. After finding a service description that matches its functional requirements according to the Chord4S service identifier, the service consumer can look over the attached quality specification. Chord4S approach supports 3 types of QoS attributes, defined as follows: Numeric, Boolean and enumerated. The syntax specification for Chord4S describes the syntax of the service, e.g., the names and data types of the input and output parameters.

2. Chord4S Service Publication

There are two traditional approaches to address the data availability issue in traditional chord approach: 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). Chord4S approach 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 the data availability. A service consumer has the opportunity to locate the functionally equivalent services from those available network nodes. Consider a Chord4S- based overlay network consisting of n network nodes, let the length of the service identifier be m and the maximum number of functionally equivalent services be k. The length of the service provider bits x should be carefully calculated to achieve even service description distribution. Obviously, a smallest virtual segment should be capable of accommodating all the functionally equivalent services, as shown in constraint (1) below:

$$2^x \ge k-1 \qquad (1)$$

Hence,

$$x \ge \log_2(k-1) \qquad (2)$$

To allocate enough bits for service provider bits, constraint (3) below should be satisfied:

$$2^x \ge k-1.\frac{2^m}{n} \qquad (3)$$

Hence,

$$x \ge \log_2(\frac{k-1}{n}.2^m) \qquad (4)$$

With constraints (2) and (4) satisfied, the descriptions of functionally equivalent services can be evenly distributed in a virtual segment which means that all of them are distributed to different successor nodes.

3. Chord4S Service Query

Chord4S approach supports 2 types of query: service- specific queries and queries with wildcard(s). A service-specific query contains complete details of the service description and is used to look up a specific service. In a system that allows four-layered function bits in the service descriptions, “Multimedia.Video.Encoder.AVI2RM” is a typical example of the service-specific query. The generation of target service identifier for the query “Multimedia.Video.Encoder.AVI2RM” is given in figure 3 as an example.

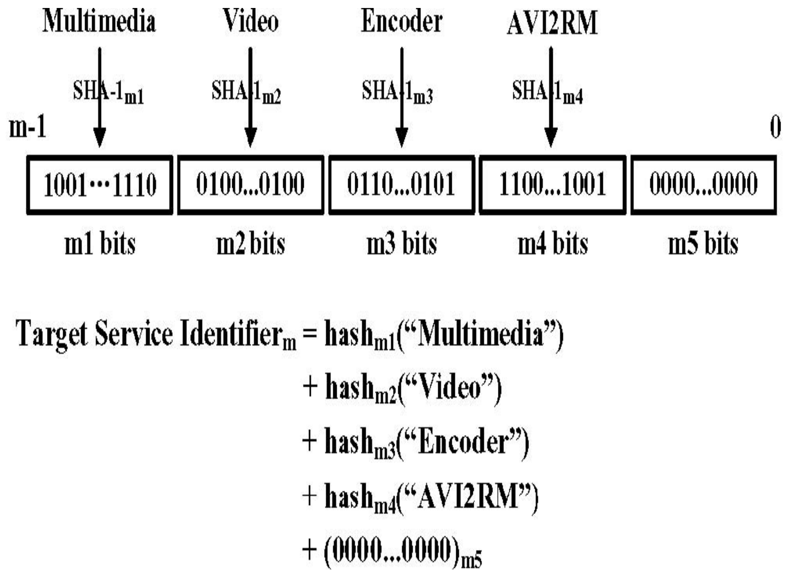


Fig. 3: Generation of target service identifier.

When solving a query with wildcards, it is actually looking up a virtual segment composed by the nodes succeeding service descriptions that fall into the target service category. The generation of the target service identifier—or more specifically target service category identifier—for a query with wildcards is similar to that for a service specific query. The difference is that the layers corresponding to the wildcards will be stuffed with 0s.

IV. CONCLUSION

Peer-to-Peer systems are based on the concept of resources localization and mutualisation in a dynamic context. In specific environment such as mobile networks, characterized by high variability and dynamicity of the network conditions and performances, where network nodes can join and leave the network dynamically, resources reliability and availability constitute a critical issue. Traditional service discovery approaches using centralized registries can easily suffer from problems such as performance bottleneck and vulnerability to failures in large scalable service networks, thus functioning abnormally. To address these issues, this paper proposes a peer-to-peer-based decentralized service discovery approach called as Chord4S. Chord4S utilizes the data distribution and lookup capabilities of the popular Chord approach to distribute and discover services in a decentralized manner. Chord4S approach supports QoS-aware service discovery and also supports service discovery with wildcards.

REFERENCES

[1] Peer-to-Peer Working Group, Bidirectional Peer-to-Peer communication with interposing Firewalls and NATs, White Paper, 2001.

[2] R. Steinmetz and K. Wehrle, peer to peer Systems and applications, (Eds) Springer LNCS 3485, 2006.

[3] 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.

[4] P. Rompothong and T. Senivongse, "A Query Federation of UDDI Registries," Proc. First Int'l Symp. Information and Comm. Technologies, pp. 561-566, 2003

[5] L. Wu, Y. He, D. Wu, and J. Cui, "A Novel Interoperable Model of Distributed UDDI," Proc Int'l Conf. Networking, Architecture, and Storage (NAS '08), pp. 153-154, 2008.

[6] 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.

[7] Qiang He, Member, IEEE, Jun Yan, Yun Yang, Ryszard Kowalczyk, and Hai Jin, Senior Member, IEEE, "A Decentralized Service Discovery Approach on Peer-to-Peer Networks", IEEE Transaction on Services Computing, VOL. 6, NO. 1, JANUARY-MARCH 2013.

[8] G. Zhou, J. Yu, R. Chen, and H. Zhang, "Scalable Web Service Discovery on P2P Overlay Network," Proc IEEE Int'l Conf. Services Computing (SCC '07), pp. 122-129, 2007.

[9] B. Sapkota, D. Roman, S.R. Kruk, and D. Fensel, "Distributed Web Service Discovery Architecture," Proc. Advanced Int'l Conf. Telecomm. and Int'l Conf. Internet and Web Applications and Services, p. 136, 2006.

[10] 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.

[11] T.H.-T. Hu and A. Seneviratne, "Autonomic Peer-to-Peer Service Directory," IEICE Trans. Information System, vol. E88-D, no. 12, pp. 2630-2639, 2005.

[12] Y. Li, F. Zou, Z. Wu, and F. Ma, "PWSD: A Scalable Web Service Discovery Architecture Based on Peer-to-Peer Overlay Network," Proc. Sixth Asia-Pacific Web Conf. Advanced Web Technologies and Applications (APWeb '04), pp. 291-300, 2004.