Volume 3, Issue 11, November 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Different Approaches for Peer-to-Peer Based Decentralized Service Discovery

Prof. D.N. Rewadkar
Associate Professor, Computer Engineering,
RMD Sinhgad School of Engineering, Pune University,
India

Yogini Bhamare
M.E. Research Scholar, Computer Engineering
RMD Sinhgad School of Engineering, Pune University,
India

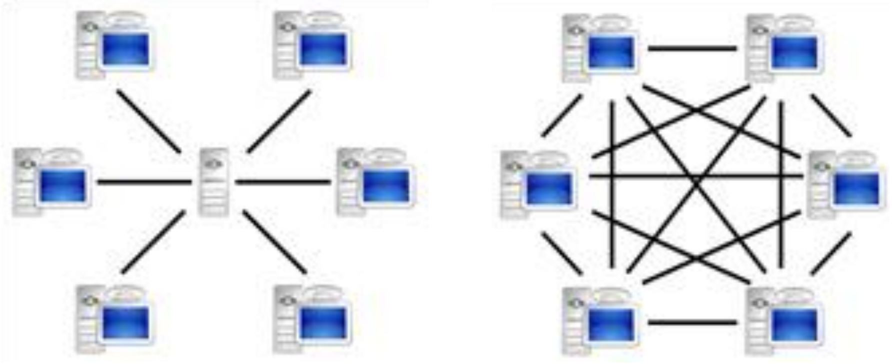
Abstract— Although Service-Oriented Computing (SOC) is emerging as a standard for developing distributed applications, having scalable, reliable, and robust service discovery mechanism is a critical issue of utilizing SOC. In traditional service discovery methods for large scalable service networks, centralized registries are used which can suffer from problems like performance bottleneck and vulnerability to failures. A peer-to-peer based decentralized service discovery approach appears to be the most natural way to address the above issues and achieve scalable, reliable and robust service discovery. For availing this decentralized service discovery approach, there should be efficient and effective method so that higher data availability can be achieved. Even though various approaches are already presented by different authors, each of the approach has its own limitations and hence this becomes the most promising area for the research. In this first step of the research, we are presenting the need to switch from centralized to decentralized service discovery approach followed by different approaches evaluated to implement peer-to-peer-based decentralized service discovery.

Keywords— Chord, DHT, DNS, Peer-to-Peer (P2P), Skip Graph, SOC, Web-Services

I. Introduction

Service computing refers to a flexible computing architecture that packages functionality as a suite of interoperable routines that can be used within multiple different systems from several business domains. Loose coupling of services with operating systems and other technologies that underlie applications is required for Service Computing. Functions are distinguished into distinct autonomous and self-describing units, or services, which developers make accessible via pre-defined interfaces over a network in order to allow users to combine and reuse them in the production of applications. These services communicate with each other by passing data in a well-defined, shared format, or by coordinating an activity between two or more services [2]. By this way, Service-Oriented Computing (SOC) is emerging as a standard for developing distributed application, but having scalable, reliable, and robust service discovery mechanism is a critical issue of utilizing SOC.

Traditional service discovery approaches of the web services technology are based on Universal Description, Discovery, and Integration (UDDI) [5]. In traditional service discovery methods for large scalable service networks, centralized registries are used which can suffer from problems like performance bottleneck and vulnerability to failures because of large number of service consumers and requests in an open SOC environment. Because of this disadvantage, applying web services in large scalable service is mostly prevented. In such a largely distributed SOC environment, to address the above issue and to achieve scalable, reliable and robust service discovery, decentralized approach seems to the most efficient way. Fig. 1 shows centralized server based and Peer-to-Peer (P2P) networks [21].



a) Centralized server based network b) Peer-to-Peer network

Fig. 1 Centralized and P2P Networks [21]

The Peer-to-Peer (P2P) technology removes centralized infrastructures to provide a universal approach for improving scalability, robustness and reliability of distributed systems. In areas such as file sharing, Voice over Internet Protocol (VoIP) and video streaming, P2P has achieved great success [1]. To leverage P2P computing and web services for improved service discovery, continuous research is going on in the SOC field. In particular, structured P2P systems such as Chord [16], CAN [11], Pastry [13], and Tapstry [18] have some characteristics that are suitable for facilitating efficient decentralized service discovery. In P2P based decentralized service discovery approach, set of distributed nodes are present which forms the P2P network. When provider registers the service, it is assigned to the relative service node for storing into the repository. Consumer can submit service query to any of the node from the network and if that node doesn't contain the required service description then it routes the query to respective node. Description of the matched query is then retrieved from the node and returned to the service consumer as a query result. This is the overall idea about how exactly service request is processed in P2P based decentralized service discovery approach. To implement such a service discovery approach DHT based and Chord-based approaches are studied by different researchers. by making use of Distributed Hashing Table (DHT), even data distribution and efficient query routing can be achieved in structured P2P systems. But in DHT based systems, descriptions of functionally equivalent services is distributed on the same successor node because hashing value is similar for these nodes. If such a node fails, any of these services will not be available to the consumer and hence DHT based P2P approaches to decentralized service discovery may not be that efficient in terms of availability of service. This disadvantage may result in serious problems in open and dynamic SOC environments where unexpected failure of nodes cannot be avoided [1]. In Chord-based approach, Chord has been used to facilitate decentralized web service discovery [6]. Emekc¸i et al. [6] present a P2P framework based on Chord for web service discovery which uses finite automata to represent web services. But these approaches are also vulnerable to the issue of data availability in open and volatile SOC environments. Descriptions of functionally equivalent services would be stored at the same successor nodes and it may lead to severe data loss in case of such node failures.

The challenge of data distribution and availability in open and dynamic SOC environment is overcome by many methods which are already presented. In this paper, we are analysing these methods along with their future road map of research. The rest of the paper gives the overview of centralized service discovery followed by different approaches evaluated to implement peer-to-peer-based decentralized service discovery. Finally conclusion is derived based on analysis of all these methods.

II. Study of Centralized Service Discovery Approaches

The centralized client/server model has been adopted for service discovery in SOC. These traditional service discovery approaches of the web services technology are based on Universal Description, Discovery, and Integration (UDDI) [5]. The UDDI Version 3.0.2 Specification [5] describes the Web services, data structures and behaviours of all instances of a UDDI registry. Going forward, as explained in [5], Web services are meaningful only if potential users may find information sufficient to permit execution of these services. The focus of Universal Description Discovery & Integration (UDDI) is the definition of a set of services supporting the description and discovery of (1) organizations, businesses and other Web services providers, (2) Web services they make available, and (3) technical interfaces which may be used to access those services. For discovering these services different centralized services approaches have been studied as discuss below.

- Extension for a query federation of UDDI registries within Web Service environment is presented in [12] by Rompothong and Senivongse. 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 a federation. But the main limitation here is authors did not provide any experimental evaluation for this.
- Wu et al. [17] 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 Domain Name System (DNS) is adopted here. Super domain servers, which are managed by a root server, are further used to maintain normal servers. This model is exposed to the same threats that DNS faces, e.g., Distributed Denial of Service (DDoS) attack, as it is based on concept of DNS.
- Eyhab Al-Masri and Qusay H. Mahmoud proposed in [4], 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 service consumers, as the crawler in the engine crawls accessible UDDI registries and gathers information in a centralized repository. As demonstrated by experimental results given, the proposed approach does improve the efficiency of web service discovery by pooling distributed information. But still there is an issue of reliability caused by single point failure.

As discussed above, in traditional centralized service discovery methods, centralized registries are used which may suffer from problems like performance bottleneck and vulnerability to failures. A peer-to-peer based decentralized service discovery approach appears to be the most natural way to address the above issues and achieve reliable, scalable and robust service discovery.

III. Study of Peer-To-Peer-based Decentralized Service Discovery Approaches

Decentralized service discovery is considered as a promising approach to addressing the problems caused by centralized infrastructures. In particular, some preliminary research has been conducted to utilize P2P computing for service discovery [1].

- Zhou et al. in [19] presents an enhanced Skip Graph, ServiceIndex, using WSDL-S as the semantic description language. Skip Graph is built by extracting semantic attributes of web services as indexing keys. It consists of a set of increasingly sparse doubly-linked lists ordered by levels starting at level 0. Fig. 2 gives a simple Skip Graph with four levels and the dashed lines show the separations between the different levels [20]. Multilayer P2P overlay network is constructed, to balance the load on peer nodes and aggregate similar indexing keys. To enhance the ServiceIndex, similar keys are inserted into the same ServiceBag. In this way, the loss of a ServiceBag will lead to the missing of all the keys in the ServiceBag, severely jeopardizing the overall availability of the keys [1].

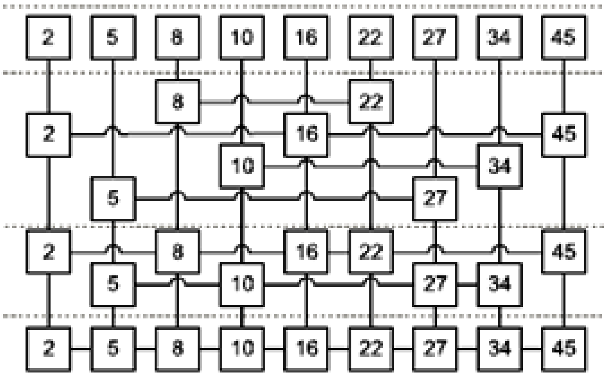


Fig. 2 Skip Graph with four levels [20]

- In [4], a multicast discovery protocol, Web Services Dynamic Discovery (WS-Discovery), to locate services on a local network is developed by Intel, Microsoft, BEP Systems, Canon, and WebMethods. In WS-Discovery protocol, request is sent to the corresponding multicast group by client to locate a target service. To scale to a large number of endpoints, protocol defines the multicast suppression behaviour if a discovery proxy is available on the network and can be switched on. Need for polling is minimized for target services that wish to be discovered by sending an announcement when they join and leave the network. WS-Discovery is becoming popular and is already being used by some software vendors, such as the "People Near Me" contact location system in Microsoft's Windows Vista operating system [1]. But WS-Discovery is specific for ad hoc networks and still there is no successful experience in applying WS-Discovery in large-scale SOC environments.
- Sapkota et al. [14] propose distributed web service discovery architecture. It is based on distributed shared space concept and intelligent search among a subset of spaces. Publishing of Web service descriptions as well as to submit requests to discover the Web service of user's interests is allowed. Integration of applications running on different resource specific devices is also supported. But in its current implementation, the shared space—the core of the architecture—is still centralized and no experimental evaluation is provided to evaluate the proposed architecture [1].
- In [6], Fatih Emekci 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 automata are used for publishing and querying the Web services within the system. Fig. 3 shows service automaton of the 'Book Seller' service (i.e., the behavior of a book selling process) [6]. Execution of Web services corresponds to a path from the start state to a final state of the service automata. The finite automata are obtained by removing loops from each path and hence it is called as Path Finite Automata through which Web Service can be executed. Also Web services ranking is provided by integrating a scalable reputation model based on sketch theory in distributed peer-to-peer framework. But the main limitation here is authors did not provide any experimental evaluation for this.

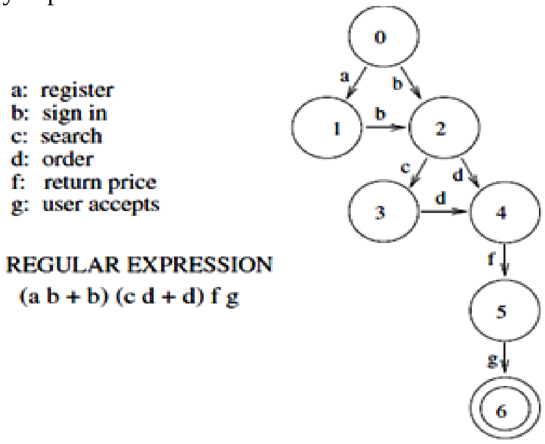


Fig. 3 An example service automaton [6]

- In [7], Hu and Seneviratne propose the approach based on the concept that service providers themselves should take the responsibility to maintain their own service descriptions in a decentralized environment. To group peer nodes by service categories to form islands on the Chord ring, decentralized service directory infrastructure is built with hashing descriptive strings into the identifiers. To handle routing across islands and within islands, Island Table and Native Table are created on every peer node respectively.
- In [9], Li et al. present PSWD, a distributed web service discovery architecture based on an extended Chord algorithm called XChord [1]. In PSWD, XML is used to describe web service descriptions and to express the service requests. To enable XML-based complicated queries, the basic P2P routing algorithm of Chord is extended with XML.
- Schmidt and Parashar in [15] describe a system that supports complex queries containing keywords, partial keywords and wildcards by implementing an Internet-scale DHT. This system assures that all existing data elements matching a query will be returned in terms of number of messages and number of nodes involved. To map the multidimensional information space to physical peers effectively, key innovation, a dimension reducing indexing scheme is used. It provides Chord having ability to perform metric-based similarity search.

Node failures would lead to severe data loss when above approaches in [6], [7], [9], [15] are adopted to provide service discovery because descriptions of functionally equivalent services would be stored at the same successor nodes.

IV. Conclusion

In this paper, we have discussed different approaches for service discovery in large scale networks along with their advantages and disadvantages. Peer-to-peer-based service discovery becomes more efficient and effective after the deficiencies of centralized service discovery are identified. Whatever existing approaches we have studied P2P-based service discovery, it is observed that none of the approach has addressed the issue of data availability in open and volatile SOC environments. To overcome this limitation, we propose the approach which would be similar to the work presented in [7] using layered service identifiers to control the distribution of service descriptions and achieve high data availability. It would support QoS-aware service discovery and service discovery with wildcard(s) along with security measures.

References

[1] 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.

[2] http://researcher.watson.ibm.com/researcher/view_pic.php?id=152.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] H.V. Jagadish, B.C. Ooi, K.-L.Y. Tan, and R. Cui Zhang, “iDistance: An Adaptive B+-Tree Based Indexing Method for Nearest Neighbor Search,” *ACM Trans. Database Systems*, vol. 30, no. 2, pp. 364-397, 2005.

[9] 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.

[10] D. Novak and P. Zezula, “M-Chord: A Scalable Distributed Similarity Search Structure,” *Proc. First Int’l Conf. Scalable Information Systems*, 2006.

[11] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A Scalable Content-Addressable Network,” *Proc. Conf. SIGCOMM*, pp. 161-172, 2001.

[12] P. Rompothong and T. Senivongse, “A Query Federation of UDDI Registries,” *Proc. First Int’l Symp. Information and Comm. Technologies*, pp. 561-566, 2003

[13] A.I.T. Rowstron and P. Druschel, “Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems,” *Proc. IFIP/ACM Int’l Conf. Distributed Systems Platforms (Middleware ’01),* pp. 329-350, 2001.

[14] 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.

[15] C. Schmidt and M. Parashar, “A Peer-to-Peer Approach to Web Service Discovery,” *World Wide Web*, vol. 7, no. 2, pp. 211-229, 2004.

[16] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications,” *Proc. ACM Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm. (SIGCOMM ’01),* pp. 149-160, 2001.

[17] 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.

[18] B.Y. Zhao, J. Kubiatowicz, and A.D. Joseph, "Tapstry: An Infrastructure for Fault-Tolerant Wide-Area Location and Routing," Computer Science Division of Univ. California, 2001.

[19] 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.

[20] M. T. Goodrich, M. J. Nelson, and J. Z. Sun, "The Rainbow Skip Graph: A Fault-Tolerant Constant-Degree Distributed Data Structure", *SODA'06 Proceedings of the seventeenth annual ACM-SIAM Symposium on Discrete algorithm,* pp. 384-393.

[21] http://en.wikipedia.org/wiki/Peer-to-peer.