A Hybrid Service Discovery Approach in Peer to Peer Systems

[1]G Jahnavai Swetha, [2]G Senthil Kumar
[1]M.Tech Student, [2]Assistant Professor
[1,2] Department of Software Engineering, SRM University , Chennai, India
[1]gunturuswetha991@gmail.com, [2]senthilkumar.g@ktr.srmuniv.ac.in

***Abstract*— Traditional Service Discovery methods are based on the centralized UDDI registries and can easily suffer from the problems of performance bottlenecks like single node failures, load balancing etc. To address these problems, many approaches based on distributed architecture have been proposed such as peer-to-peer-based decentralized service discovery approaches (CAN, Chord,Chord4S etc)which suffer from routing inefficiency and causes over head delays. Hence a new hybrid approach that is based on the decentralized peer to peer system as well as a centralized global repository is defined to address the problems. Based on the Chord 4S it supports service query with wildcards also. Load Balancing and Data Availability are further extended by distributing the domain specific services with respective peers and also a method called Request Mediator is introduced to reduce the overhead delay and improve the performance of the system.**

***Index Terms*— Peer to Peer System, Service Query, SOAP, UDDI, Web Services, XML**

I. INTRODUCTION

Web services are the services that are made available on the internet from a web server for the web users or the other web connected programs .They are application components which are in the XML format[4]. Web services can be used by any application irrespective of platform in which it is developed. Web service description is provided in WSDL [3] document. It can be accessed from internet using SOAP [2] protocol. In industry, many applications are built by calling different web services available on internet. These applications are highly dependent on discovering correct and efficient web service. The discovered web service must match with the input, output, preconditions and effects specified by the user. UDDI stands for Universal Description Discovery and Integration[1] .UDDI is the mechanism for registering and discovering web services. It is a platform-independent framework for describing services, discovering businesses, and integrating business services by using the Internet. Before the introduction of UDDI into the web , there was no Internet standard for businesses to reach their customers and partners with information about their products and services. Search process in UDDI is based on keyword matching mechanism.

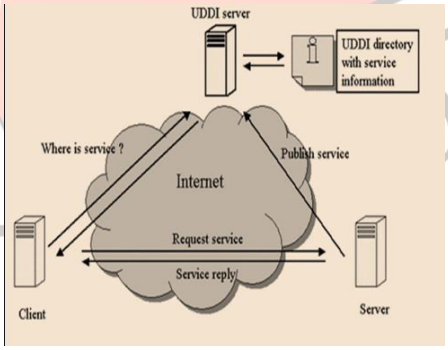

Figure 1. Service Lookup using UDDI

An example scenario to illustrate the working of UDDI

For flight rate checking and reservation the industry publishes an UDDI standard, then airlines can register their services into an UDDI directory. Travel agencies can then search the UDDI directory to find the airline's reservation interface. When the interface is found, the travel agency can communicate with the service immediately because it uses a well-defined reservation interface.

Service Discovery Approach

Service Discovery is a process of obtaining a set of services that can possibly fulfill the user requests. A web service discovery process is performed in three major steps. First step is advertisement of web service by developers. Providers advertise web services in public repositories by registering their web services using web service description file written in WSDL. Second step is web service request by user. User sends web service request by specifying the requirement in predefined format to web service repository. Web service matcher which is core part of web service discovery model, matches user request with available web services and finds a set of web service candidates. Final step is selection and invocation of one of the retrieved web services. Discovery of correct web service depends on how mature web service matching process is. i.e.; how actual requirements of user

are represented in formalized way and how they are matched with available services

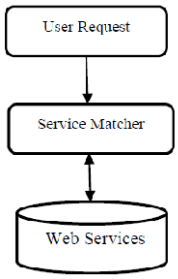

Figure 2. Service Discovery

II.RELATED WORK

Traditional service discovery approaches of the web services are based on Universal Description, Discovery, and Integration (UDDI).Many approaches for the architectures of UDDI have been proposed to improve the performance but still suffer from performance bottlenecks.

A. CENTRALIZED ARCHITECTURE

In the traditional system, UDDI [10] acts as the centralized server and have centralized registries to store services.As a result of these centralized service registries used, the UDDI may easily suffer from problems such as performance bottleneck and vulnerability to failures as the number of service consumers and requests increase. This inherent disadvantage prevents web services from being applied in large scalable service networks. There are many limitations for the centralized architecture like, Firstly ,if the UDDI server fails it caused single node failure .Secondly, when there many number of service requests it is very difficult for a single server to handle and effects the scalability of the system. centralized infrastructures inherently suffer from poor performance in an open SOC environment that demands high scalability. Measurements have been taken to tackle the problem by employing distributed UDDI registries. In [5], Rompothong and Senivongse propose a federation of UDDI registries to enlarge the search space for service queries. Although a UDDI Federation Agent is added as an extension to a standard UDDI registry to forward queries to other federating nodes, the authors did not provide any experimental evaluation. A Web Service Crawler Engine is proposed to address the performance issue caused by employing an enormous number of UDDI registries [6]. The engine crawls accessible UDDI registries and collects information in a centralized repository via which service consumers can efficiently discover required web services. The proposed approach does improve the efficiency of web service discovery by pooling distributed information, as demonstrated by provided experimental results. However, it jumps back to the issue of reliability caused by singlenode failure.

B.DECENTRALIZED ARCHITECTURE

In a decentralized Architecture, the information or the services are distributed across the UDDI registries .A UDDI registry can be referred to be a node. This architecture overcomes the problems of the single node failure and improves the registry performance but yet lack in load balancing. Decentralized service discovery is considered as a promising approach to addressing the problems caused by centralized infrastructures. Many approaches have been proposed based on the peer to peer technology[8] .CAN[7] is proposed in which the hash tables are used to map the keys to values and improves the system but yet suffers from problems like ,flooding on every request is clearly not scalable, flooding may fail to find the content in the system. This makes the system vulnerable and expensive. Schmidt and Manish proposed a paper on Indexed Peer to Peer Technology[9] that supports large scale decentralized, real-time search capabilities in which complex queries with partial keywords and wildcards are supported but the query cannot be efficiently routed to the nodes .Hence, suffers from the routing inefficiency. Replication of the data has been implemented to the Chord [11] to improve the load balancing and low latency. In an open SOC environment, replication and redundancy both have disadvantages. The replication approach leads to sophisticated maintenance for data availability. The redundancy approach requires significant change to the original service descriptions which may not be acceptable by the service providers. Both approaches may result in a considerably large burden on the system. Later Chord4S[12] is implemented which is a highly scalable and adaptable to robust environments .But this extended peer to peer approach has to encounter the overhead delays in routing.

III. PROPOSED METHODOLOGY

The proposed Method is a Hybrid Approach supports high data availability along with the load balancing issues.It also further extends the scalability of the system.

Hybrid service discovery approach

In this proposed system there is decentralized Peer to Peer Architecture along with the centralized global repository. Decentralized method represent as each node has its own local registry act as Domain based repository and it will able to store a Domain specific services in local registry, the whole system is managed by one global centralized registry is done centrally. Request mediator will add various service consumer request to peer capacity list , which is based on priority methods in local registry. Non domain specific repository and backup of domain specific repository are available in global repository.

Network establishment

A peer to peer network is established by creating individual nodes and a local registry to each of them.The minimum size of nodes has to be given.Each peer contains the specific services to a particular domain.

Service creation

Each time a new service is created ,it is stored in specific peer with respect to the domain .Each service contains service description and service specification .These details are stored in the specific peer.All these domain specific services respective to peer are stored in the global repository along with the peer node information.

Service query

In service query the user can know all the services available in the network. Then the user can select any one of the service from that. A service-specific query contains complete details of a service description and is used to look up a specific service. Based on the clients requirements the service query is classified of two types[11],namely,

- *Service Specific Query (SSQ)* : A service-specific query which contains complete details of a service description and is used to look up a specific service. The objective of using service specific query is usually to look up a group of functionally equivalent services provided by different service providers.
  Example : Bank ->Andhra Pradesh->UBI

- *Wildcard Query* : A wildcard contains various sub domains in it. Sometimes service consumers need to search for categories of services but not just one. Hence a wildcard query gives all the services related to that domain.
  Example : Bank

Service distribution

After the client request is given in the form of the service query,Request Mediator identifies the peer that has the requested service and adds the request to the peer capacity list . then if the peer is busy for long time it is retrieved from the global repository.And then the client gets the service response from the global repository.

IV.EXPERIMENTAL SETUP

This section describes the implementation details of execution of the project. To validate the Hybrid Service Discovery Approach a minimum set of 8 peers are taken in this case study. Service Provider need to register their services in the peer nodes of the peer to peer system.Each peer contains its domain specific service.So the registered service automatically gets stored in the respective peer. A backup of all the services is maintained in the global repository which also maintains the peer node information. Here , 8 different domain specific web services are considered like Automobile, Bank, Book, Cement Company, Insurance Company, IT Company, Telecommunications Company are taken which are XML files with various service descriptions and service contents respectively . The Request Mediator transfers the service request either to the peer nodes or the global repository based on the priority methods. Here, the priority is based on the load balancing factor,i-e, the peer is assumed to be busy if its load balancing capacity is high and then the request is transferred to the global repository inorder to reduce the waiting time for the client.Load balancing capacity is measured by the Swap_Space method in memory bytes.Swap_space method gives which peer is using more amount of space in the server currently and is assumed to be busy handling many service requests.The below figure 3 illustrates the functional requirements of the system.

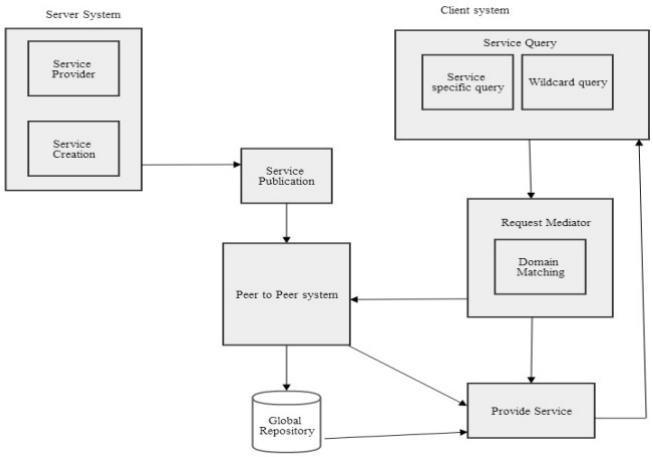

Figure 3. Experimental setup of the system

Client send his requests in two types like the service specific query and wildcard query. The datasets taken are,

CLIENT	DATASETS	
Service Query	Service Specific Query	Wildcard Query
	1.Service Name	Service Name
	2.Service Description	

Figure 4.Datasets for query evaluation

V. EXPERIMENTAL RESULTS

This section examines the performance of the peer to peer system. When a client requests for a service ,then the service descriptions of the service are returned. The main aim is to overcome the bottlenecks of the centralized system as well as to improve the load balancing.

Single node failure

Single node failure occurs when failure of a single node in the system results in the failure of the whole system.The Hybrid Service Discovery Approach successfully overcomes the bottleneck issues of single node failure .Here firstly,service descriptions of the various domains are stored in each peer respectively.Incase if a peer containing a particular service is down the service can be retrieved from the Global Repository .

Load Balancing

Here when the service request is made by the client the peer containing the specific service is identified by domain analysis and handles the service request. Then the peer is checked if it is busy by the Request Mediator that calculates the current workload of the peer. Incase if the workload of the peer is higher than the threshold value it is assumed to be busy and the requested service can be fetched from the Global Repository.

Response Time

Response time is the time from which the server sends a response message until the message reaches its targeted client. It is computed by dividing the total delay experienced by the number of clients.

Data Availability

A unique feature, and also a main design goal of Hybrid Service Discovery approach is the high data availability. Here along with the peer nodes the Global repository contains the service descriptions. Hence, if any peer node fails still the service descriptions can be fetched from Global repository.

VI. CONCLUSION AND FUTURE WORK

Service discovery is an important factor in the Web Services. Many decentralized approaches based on the Peer to Peer System have been proposed to overcome the performance bottlenecks of the centralized systems but yet lack in some of the issues like overhead delay, flooding of input requests. Hence, a Hybrid Service Discovery approach is proposed in this paper that supports a decentralized peer to peer architecture in which the service information is distributed to the respective nodes .It also supports a centralized global repository that contains all the peer node information. This feature improves the performance of the system by overcoming single node failures and further increasing the data availability of the system. The service request is transferred to the global repository when the peer node is busy. This improves the load balancing and improves the scalability and efficiency of the system.

In future the semantic information can be integrated to the existing system and semantic service discovery can be implemented so as to improve the flexibility and accuracy of the system.

REFERENCES

[1] "Universal Standard Products and Services Classification(UNSPSC)," http://www.unspsc.org/, 2012.
[2] Word Wide Web Consortium, SOAP Version 1.2, W3C recommendation, second ed., http://www.w3.org/TR/soap, Feb. 2010.
[3] E.Christensen,F.Curbera,G.Meredith,S.Weerawarana,"Web Services Description Language (WSDL) 1.1", W3C Note, 2001.
[4] "WebServicesArchitecture", http://www.w3.org/TR/ws-arch/
[5] P. Rompothong and T. Senivongse, "A Query Federation of UDDI Registries," Proc. First Int'l Symp. Information and Comm. Technologies, pp. 561-566, 2003 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.
[6] E. Al-Masri and Q.H. Mahmoud, "Crawling Multiple UDDIBusiness Registries," Proc. 16th Int'l Conf. World Wide Web (WWW '07), pp. 1255-1256, 2007.
[7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," Proc. Conf. SIGCOMM,pp. 161-172, 2001.
[8] 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.
[9] V. Gopalakrishnan, B.D. Silaghi, B. Bhattacharjee, and P.J. Keleher, "Adaptive Replication in Peer-to-Peer Systems," Proc. 24th Int'l Conf. Distributed Computing Systems (ICDCS '04), pp. 360-369, 2004.
[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] Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications Applications," Proc. ACM Conf. Applications, Technologies, Architectures,and Protocols for Computer Comm. (SIGCOMM '01), pp. 149-160, 2001.
[12] Qiang He, Member, IEEE, Jun Yan, Yun Yang, Ryszard Kowalczyk, and Hai Jin, Senior Member, IEEE, "A Service Discovery Approach on Peer-to-Peer Networks" *IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 6, NO. 1, JANUARY-MARCH 2013.*