International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 1, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

A New Model for Distributed Development Registry in Web Service Discovery

Somayyeh Ehteshami
Dep. of Computer Engineering & Information Technology,
Islamic azad University, Saveh Branch, Markazi, Iran.
email: Ehteshami@gmail.com

Mohammad Reza Heydari Nejad
Dep. of Computer Engineering & Information Technology,
Payame Noor University, PO BOX 19395 - 3697, Tehran, Iran.
email:Mrheydarinezhad@pnu.ac.ir

Abstract: **The key technology for search servers and dispersion in web services is UDDI (Universal Description, Discovery and Integration) which can be used for dispersion, discovery, and integration of web services with the help of WSDL. Without UDDI, the web service providers will fail to distribute service information, and the users cannot do the search. They also are not able to remotely connect to their required services. Most of the current UDDI models are centralized, so that if with existence a great number of services to register or to receive queries, their performance will be reduced. To obviate the problems with centralized registry, different models of distributed registries have been presented. In this study, we attempted to enhance the performance of one of the distributed service stores models. It takes time to make sure that the service information in registries is accessible and useful. The consuming time for the accessibility of the service depends on a strong relationship between UDDI and the service providers. This is while no method is available to check the real accessibility of the services. Hence, the designed model can increase the validity of the service available at service stores to a tangible extent. Moreover, in the proposed protocol, a method will be presented for reducing the network traffic which will be later implemented.**

Keyword: **Web Service, UDDI Service Stores, Service Discovery, Distributed Systems, P2P Architecture**

1. INTRODUCTION

Web services provide an appropriate technical foundation for cooperative and collective capability of programs. The service provider describes web services by WSDL in order to make web services available for the user, and also he publishes descriptions in the service registry. Hence, the service requestor may be able to find them (the web services) easily [1].

This paper aims at presenting a proposed solution to improve service discovery which uses the merits of P2P methods and provides a model for improvement of web services search speed. Moreover, since in all centralized UDDI models, collects service information inactively (passively), i.e. inactively waits for service dispersion, update or discovery request; therefore, the real time of service information validity is not guaranteed[2]. In UDDI distributed models, it takes time to be assured about the accessibility and efficiency of service information within registries. By the way, the service's consumed accessibility time depends on a strong and advanced relationship between UDDI and service providers. There is no method to check for the actual accessibility of services [4]. Hence, we have designed a model which is able to enhance the service descriptions available in service stores to a tangible extent.

In an article written in 2010, a P2P model called PDUSE was employed for service stores' release [5]. This model was designed as a loop model benefited from the merits of Chord protocol which saves service indices in tallies and presents a comprehensible and flexible model for distribution and discovery of services. Chord is one of the most well-known P2P systems based on distributed hash table (DHT), and of its most significant properties are scalability, load balancing, high performance, simplicity, etc. This simple structure and efficient algorithm, has a high compatibility for data locating along with error tolerance, which is being applied extensively for information storage and query. Chord is a logarithmic operation for tracking the keys according to the nodes and guarantees the recovery of any stored entry within the network.

This protocol receives an m-bit ID as an input and returns the node which has stored the corresponding value of the key. Data locating can be easily performed by attributing one key to each value and restoring the "key-value" by which the key is mapped. When the nodes join the system or leave it,the protocol efficiently recovers itself; and even if the system is constantly changing, it can still respond to queries.

Despite these favorable properties, Chord protocol has some demerits as well; one of its demerits is that within this protocol none of the existing network nodes does not consider the nodes in their geographical domain; that is each node directs the search request based on its own routing chart, instead of searching for the intended information and data first within the nodes of its geographical domain. This causes the search time to be increased and the network's bandwidth to be wasted. In this research study, our purpose is to improve the Chord protocol by adding a new capability to it, so that it can also consider for the physical location of the nodes present in the network. This new capability does not disrupt the general procedure of search in Chord.

Specifically, distributed UDDI only stores the index for web services in PDUS, while descriptions of web services (like WSDL document) still stores at the service providers' side [3]. On the one hand, it reduces the pressure on information storage and transfer in distributed UDDI. On the other hand, the requestors are able to determine whether services are at hand before downloading the WDSL files.

The structure of service index can be described as the Figure 1.

Address of Backup Index	Address of Service Document (such as WSDL)	Service Description	Service Category	Name of Service	Service ID

Fig.1. The Structure of Service Index [4]

2. The Proposed Architecture

In our proposed model, we made use of Chord protocol [6] because of its capabilities, flexibility, error tolerance and high performance, simplicity, limited number of measurement in accordance with the logarithm of number of nodes during the search for key, high scalability and flexibility regarding the transmission of nodes in and out of the network.

Two points have been considered regarding the model:

1. Since information validity is an important part of registries, and regarding the low level of updating for distribution of information across registries, a particular mechanism is needed to automatically monitor and check the condition of the service and information updating within the registries.
2. Despite having numerous capabilities, Chord protocol demonstrates a low performance regarding routing across nodes; because it neglects the physical network's topology during the entrance of new nodes to the system.

To solve the mentioned problems, we suggest a two-layered model called NMChord which has obviated the discussed parameters and the problems with previous models. In this model, the first layer of service stores is distributed into the Chord loop. Each node is called RNC(Register Center Node) to be better understood. The same mechanism of Chord protocol is used for service discovery and service storage. Below the stages for service discovery and the internal structure of RNCs are expanded.

The second layer is for updating and validity check of the services; which has been added to the system in order to improve validity of services and performing the automatic update. Each node from the second layer is called NM.

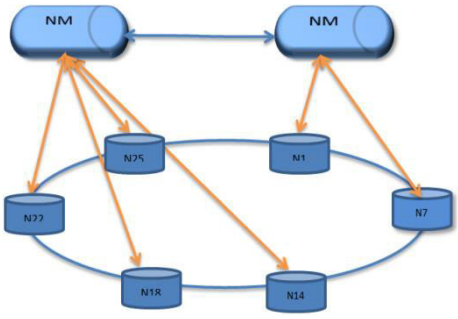

Fig.2. The Architecture of NMChord Proposed Model

3. Information Validity

The service information validity in registries is very important. Regarding the low organizational update of information release in the registry, a particular mechanism has to monitor and check the condition of the service and automatically update the information in registries. In this design, a registry service called NM (Node Monitoring) server checks the real time condition of the services and collects the service information regularly at regular periodic intervals. To facilitate the setup process and reducing the loading, only the name of the service, the key and the version of it are stored in NM service store.

The Figure 3 is the internal structure of NM.

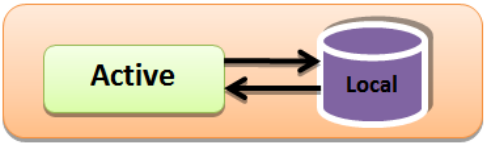

Fig.3. TheInternal Structure of NM

NM Structure

This part is only responsible for updating and warranting the validity of service information. In NM's information store, only the name of the service, the key, and the service release are kept to account for simplicity and reducing the traffic load. Each NM node is considered as a main partin updating and warranting the validity of RNC stores. Considering the fact that each NM is implemented for updating, managing and monitoring, a Scope Location is defined in each implementedNM in which the NM geographical domain is specified. The main function of each node is to check for functionality and update of RCN. NM works the same as load balancing, i.e. the nodes of the defined geographical domain will occur at the time of uploading. This way no additional load will be established on NM.

Scope Location receives information from service stores' location chart.

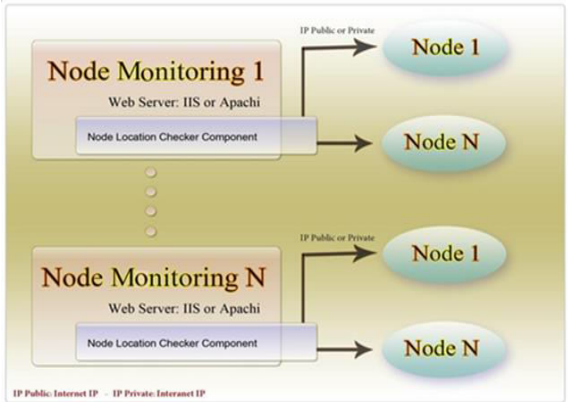

Fig.4. NM Technical Model

Updating Mechanism of NM

Monitoring and updating pallet sends a connection signal to the service providers in its ownstore. Upon receiving any response from the provider, the service release, the name and the key of the service are compared to the existing information at the side of the service provider. If there is any difference, the monitoring pallet enters into the updating phase, the updated information are

sent to NM server, and the service index available at the local store and the cache store are updating by the manager of service stores. In case of no difference, a Non Update message will be sent to the NM server.

If no response and connection from the side of the service provider is seen, using a single TTL number, the NM server sends a connection request in certain time intervals and up to ten times for the service provider. If the provider does not respond, the service information is announced invalid and the data for local and cache stores will be updated through information retrieved from the manager of caches in the lower layer. Therefore, service search and discovery are only performed across valid services.

Among the advantages of NM model we can refer to service information updating and service validity enhancing. In addition, since each NM manages the service information of geographically closest caches, the information updating traffic reduces to a significant degree.

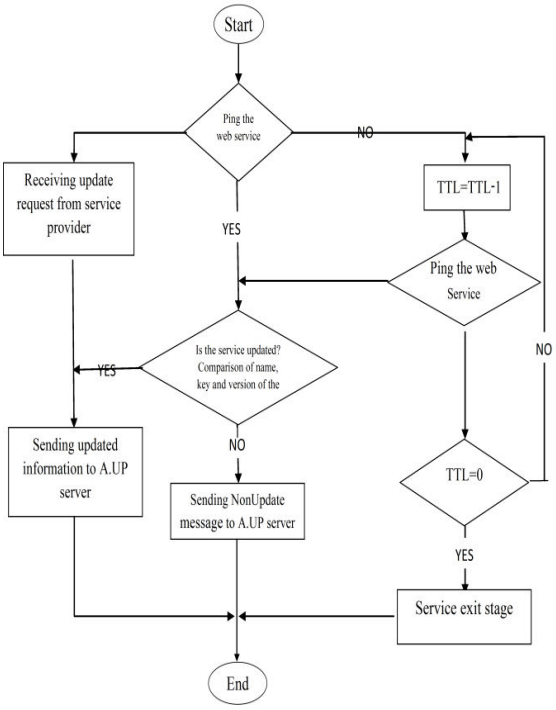

Fig.5. The Service Information Update

4. Adding an RCN Node to the Loop

In Chord protocol, each node contacts to a service node at the time of entering into the system. This central server keeps the current status records of the network, including the number of present peers, the IP address, the port for each peer, and the order of loop nodes. If a node wants to join the network, it sends a joining request to the server. According to the information it has of the network and with regard to the new node's ID, the server sends a message to the new node of the IP address of the node before which it should be placed.

Our recommended solution is to change the server in a way that besides keeping the previously mentioned information, it will store the physical location of network nodes in itself. This happens during the node's joining process with the network.

Each new node should attach its geographical domain to the joining request and send it to the central server. As soon as the request message is delivered, the central server has to send a response message containing the IP address of the successor node, and also one of the nodes of the same geographical domain to the new node. After complete attaching to the network, the new node makes a contact with a node of its own geographical domain to fill the location chart.

Since the network is constantly changing, the available nodes in each geographical domain should communicate with each other on specified time periods; in other words, they should ping each other. Some nodes join the network and others leave it; thus, the location store charts should be updated in order to route the requests correctly. Therefore, the location chart contains the names of the nodes which are close to the current node.

Since the manager of service stores information has a direct connection with NM, every NM has access to the information of location charts of all nodes, including the location of those nodes which are close together. By registering the new node in the Chord loop, it simultaneously is registered in the related NM to be available for the updating operation.

5. Web Service Discovery in the Proposed Model

During the process of service discovery in this model, the request is delivered to the service manager after the requestor requests service via Web GUI. Then, the request arrives at the enquiry processing unit through the service manager. There, the request's keywords are extracted, and the key values are extracted in DHT environment according to the keywords through keyword outlining unit. The query processor will be localized in the store. If found, it gives the service index back to the service requestor; otherwise, the route processor searches in the cache chart. In case of finding the given service index in the cache chart, it is returned to the service requestor and finally, if the given service is not found in the current node and the recent queries saved in the cache chart, the request will be directed to the routing chart.

The routing processor uses the information from the routing and physical location charts to select the best next node; so that traffic in the network and extra costs will be reduced as much as possible, and the closest nominated node will be selected. The route chart is constituted of RCNs' address information for service queries, and the physical location chart includes RNCs' the RCNs' physical location information. The information of these two charts is updated when a new node is registered and added to the loop.

To run the Chord model, it should be first implemented by a ChordNode architecture programming language. In

International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 1, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

this section, Net platform with Net Remoting capability has been used for implementation of ChordNode.

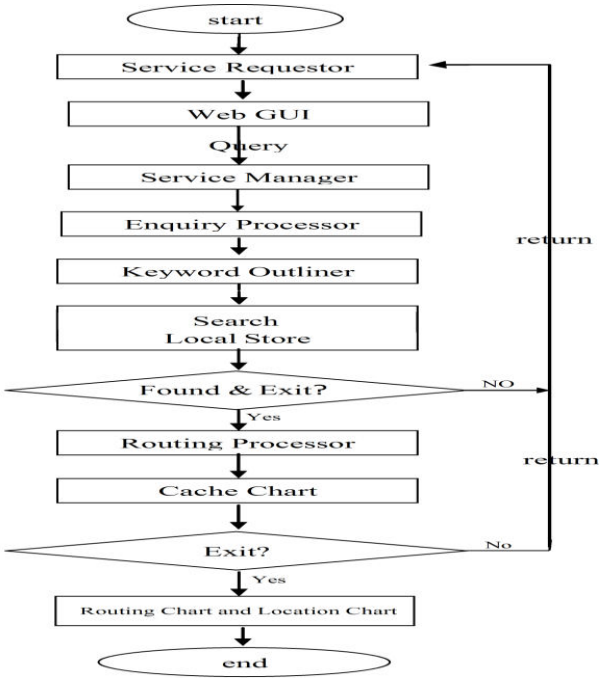

Fig.6. Service Discovery

A 3-layered model is defined in the performed implementation. The details for each layer are as follow:

1. NChord Lib which is the architectural implementation library of Chord Node and Chord Instance.
2. NChord Server via which the activation and Performance of Chord and Remoting are acheived.
3. 3. Node Monitoring implementation and RCN service store updating layer via Net Remoting service update.

6. Experiment Results

To evaluate the design of NM Chord, we make use of the following criteria:

- Degree of availability: as a ratio of requests which are found to be successful. In a real B2B environment, service requestor tends to use service information directly from the service registry. Therefore, lack of validity for the discovered service information is a significant reason for the failure of functional B2B programs.
- Average response time: as the average elapsed time, from distributing the query until finding a desirable and accessible service.
- Total traffic costs: as the traffic incurred by the enquiry, up to the time of receiving a response.

In this section, the proposed simulated model and the results are compared with the existing assessments of the previous models (3). As the proposed model is implemented in the Net Framework environment, Matlab software was used for the preparation of graphical outputs due to the limitation of C#.net environment. Therefore, simulation of the model was performed in such an environment. Our simulation was done through a network consisted of 100 nodes, at 1000-node increase, and 100 times repetition per minute. Different tests of 30, 50, 100, and 1000 times repetition eventuated very close results. Due to the time limitations, 100 times repetition per minute was selected. In every repetition, the scale of 1000 service indices was used as the stores' input.

In the Figure 7, the obtained values from the simulation for each of the studied parameters are given.

In the following section, we compare the previous model with the obtained results from the stimulation of the proposed model.

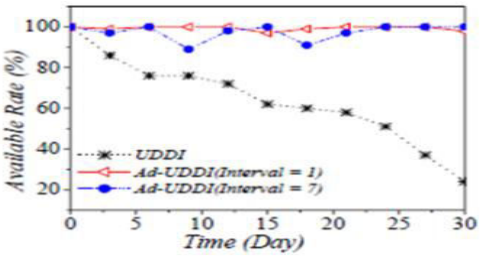

Fig.7. (A)

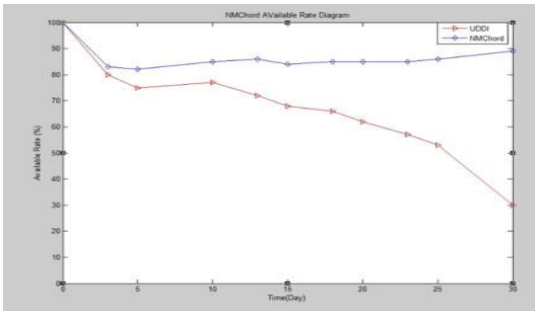

Fig.7. (B)

Fig.7. Comparison of Accessibility Between the Existing Model (A) and the Proposed Model (B)

Comparing the performance of the proposed system with that of the previous AD-UDDI model in terms of accessibility parameter, the current model demonstrates a significant improvement in comparison to the centralized UDDI [7], and has similar results as the existing model.

The next parameter is the average response time. The results are shown in the figure below. It is obvious from the obtained results that this parameter is better than its counterpart in the centralized UDDI model; but in comparison, AD-UDDI model has better results.

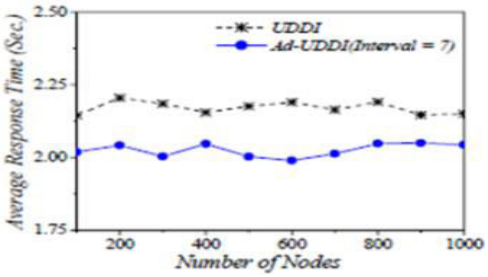

Fig.8. (A)

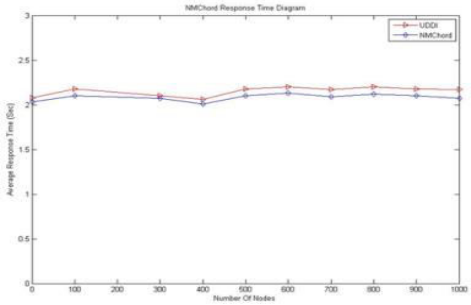

Fig.8. (B)

Fig.8. Comparison of Response Time Parameter in the Existing Model (A) and the Proposed Model (B)

The last parameter which was studied was traffic costs, which in the proposed model is seen to have improved greatly in comparison to the previous model; because it made use of the facilities from the location chart. This amount of improvement is a result of sending enquiries to the nodes of close physical distance, which in turn leads to a decrease in network traffic and in consuming bandwidth.

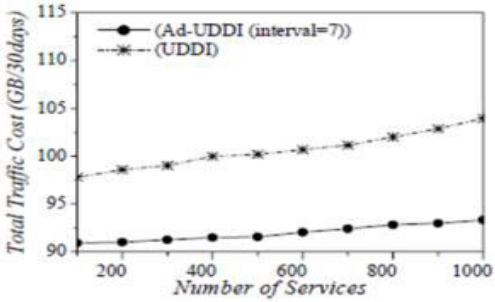

Fig.9. (A)

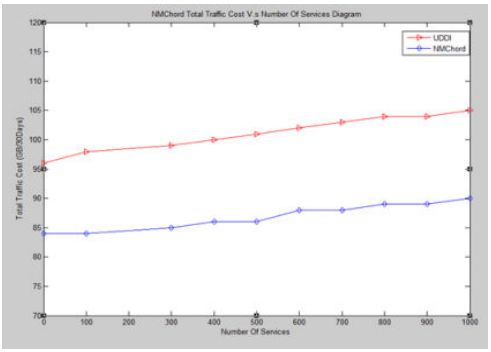

Fig.9. (B)

Fig.9. Comparison of Traffic Costs in the Existing Model (A) and the Proposed Model (B)

7. Conclusion

Despite desirable properties for implementation of P2P networks, Chord network does not consider network topology, and thus its performed queries do not have good query time and the optimal route of the queries is not accomplished. On the other hand, in the model presented in the previous research, there is no mechanism to automatically update the service information available at service stores. This leads to the decrease in the degree of validity for service information. Therefore, the model presented in this research study suggests a solution for this problem and utilizes the merits of automatic updating of service information in PDUS P2P loop model. The results obtained from the simulation show relative and desirable improvement of the current model.

All available web services are distributed as indices in RCNs. The service index makes the system to be able to show better efficiency in the discovery and distribution of web services. This model also saves the real location information of service indices' storing nodes in the physical location chart when they are joining with the system and the Chord loop. This results in the enhancement of service discovery performance and the reduction of network traffic. This issue makes a big contribution to the flexible and simple structure of Chord protocol, in which no attention is given to the physical location of the existing nodes in the loop, which leads to the waste of network costs. Moreover, the current structure has not considered any mechanism with regard to storing service indices used for updating service stores. In the current study, adding a new layer, nodes called NM are responsible for updating the information of service indices and enhancing the validity of the current services.

For a better performance, it also uses information from the physical location chart, so that each node is responsible for updating numerous physical nodes which are located close together in the Chord loop. The presented model aims at improving the performance of service discovery, keeping the service stores up to date, and enhancing the validity of the investigating services.

References

[1] Chinnici, Roberto and Moreau, Jean-Jacques and Ryman, Arthur and Weerawarana, Sanjiva, “Web Services Description Language (WSDL) version 2.0 part 1: core language”, WWW Consortium,vol. 26, pp. 19-28, 2007.

[2] D'Mello, Demian Antony and Ananthanarayana, VS. “A Review of Dynamic Web Service Description and Discovery Techniques.” Integrated Intelligent Computing (ICIIC), First International Conference on, pp. 246-251, 2010.

[3] Du, Zongxia and Huai, Jinpeng and Liu, Yunhao, “Ad-UDDI: An Active and Distributed Service Registry”, Technologies for E-Services. Springer, pp.58-71,2006.

[4] R.Rajmohan ,N.Padmapriya ,S.K.V.Jayakumar , “A Survey on Problems in Distributed UDDI”, International Journal of Computer Applications, vol. 36, no.3, pp. 887-975, 2011.

[5] Yulin, Ni and Huayou, Si and Weiping, Li and Zhong, Chen, “PDUS: P2P-based Distributed UDDI Service Discovery Approach”, IEEE International Conference on Service Sciences (ICSS), pp. 3-8. 2010.

[6] Stoica, Ion and Morris, Robert and Karger, David and Kaashoek, M Frans and Balakrishnan, Hari. “Chord: A scalable peer-to-peer lookup service for internet applications”. ACM SIGCOMM Computer Communication Review, vol. 31, no. 4, pp. 149-160, 2001.

[7] Clement, Luc and Hately, Andrew and von Riegen, Claus and Rogers, Tony and others. “UDDI Version 3.0. 2, UDDI Spec Technical Committee Draft”, OASIS UDDI Spec TC, 2004.